2

REPORT DOCUMENTATION PAGE

AD-A242 206

DTIC ELECTE NOV 1991 D C

1b. RESTRICTIVE MARKINGS	
3. DISTRIBUTION/AVAILABILITY OF REPORT	Unclassified Distribution Unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	
5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION	Clemson University Clemson Apparel Research
6b. OFFICE SYMBOL (If applicable)	
7a. NAME OF MONITORING ORGANIZATION	Defense Personnel Support Center
6c. ADDRESS (City, State, and ZIP Code)	500 Lebanon Road Pendleton, SC 29670
7b. ADDRESS (City, State, and ZIP Code)	2800 South 20th Street P. O. Box 8419 Philadelphia, PA 19101-8419
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	Defense Logistics Agency
8b. OFFICE SYMBOL (If applicable)	
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	DLA900-87-D-0017 DO 0005
8c. ADDRESS (City, State, and ZIP Code)	Room 4B195 Cameron Station Alexandria, VA 22304-6100

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
78011S			

11. TITLE (Include Security Classification)

"Production Planning for Quick Response"

12. PERSONAL AUTHOR(S)

Dr. J. J. Kanet and Dr. J.S. Davis

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
final	FROM 2/17/89 TO 8/30/91	91/10/10	83

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	
11	05		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The concept of quick response through linkage has rightfully generated considerable discussion, but to date there has been little action in terms of the scientific design of techniques to implement it. Attempting to force-fit MRP-based planning methods to this problem appears ill-advised and clearly not consistent with the direction in technology towards networks of microcomputers. Our objective in this project was to develop and evaluat several competing approaches to production planning which, in contrast to MRP-based approaches, simultaneously determine both a production schedule and material plan. We conducted the study by developing various scheduling algorithms and implementing them into a prototype interactive scheduling system.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS		
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL
Dr. Christine W. Jarvis	(803) 646-8454	

PRODUCTION PLANNING FOR QUICK RESPONSE
(DLA900-87-D-0017)

J. J. Kanet
J. S. Davis

Clemson University, Clemson, South Carolina

October 1991

CONTENTS

ABSTRACT

OBJECTIVE

The objective of this research project was two-fold: first, to develop good algorithms for solving the type of scheduling problem as specified below; and, second, to investigate state-of-the-art software tools for the development of scheduling systems that deploy such algorithms.

THE PROBLEM

The specific problem addressed in this project was the discrete version of the well-known single workcenter economic lot scheduling problem. The specification here differed from the usual formulation in a number of important ways: demand for a set of different products is assumed to occur in discrete "batches" over an extended planning horizon; the setup times for the products are sequence-dependent; early shipment of orders is forbidden; batch splitting is permitted; and the penalty function is of a general convex form allowing penalties for both early and late order completion.

METHODS

Effort associated with algorithm development entailed: reviewing existing methods for solving similar combinatorial scheduling problems; applying the more notable approaches to the problem at hand; implementing the algorithm(s) on a computer; and performing comparative algorithmic studies over a well-designed set of sample problems. Effort associated with testing the viability of software tools involved developing a production scheduling system prototype called the Clemson Quick Response Planner. The Quick Response Planner is suitable for demonstration to and evaluation by both apparel manufacturers and developers of production planning software.

APPLICATION

The problem specified here has broad application to the apparel industry. It applies to any situation in which a resource (an important piece of equipment, a worker, etc.) is in short supply and must be carefully scheduled in order to complete all orders satisfactorily. Examples include scheduling of cut orders which must be entered into the cutting department, scheduling of individual workers or work centers, or the scheduling of Unit Production Systems.

I. INTRODUCTION

SOUND PRODUCTION PLANNING: PREREQUISITE FOR QUICK RESPONSE

It has been estimated that unnecessarily long response time is costing the U.S. textile-apparel-retail complex over $12.5 billion every year. (See "The Home Team Advantage -- Timely Response Through Technology and Cooperation," 1986 Apparel Research Conference program.) Clearly, the future health of the U.S. apparel industry will require responsive, coordinated interaction among all members of the apparel network: retailers, designers, apparel manufacturers, and textile manufacturers. Recognizing this need, the U.S. apparel industry is currently hard at work developing electronic "linkage" among companies. Indeed, the American Apparel Manufacturers Association has established a textile and apparel linkage council (TALC) with the express charge to promote linkage and develop industry standards. Linkage is an important component of the overall industry effect now popularly known as "quick response."

Improved linkage will expedite communication, but will increase the demands on individual companies to use this information more effectively for production planning. Each organization in the industry can be thought to be participating in a network of cooperative enterprises. Based on the requirements of its own customers, each enterprise must be able to quickly plan and replan its production and the associated requirements for materials from its suppliers. The responsiveness of the industry will only be as good as the planning capability of the weakest member in the network.

The clear trend today is that apparel companies must increasingly be able to produce a variety of garments with short lead times. Current Material Requirements Planning (MRP) methods for manufacturing planning are unable to adequately react to the dynamics of the apparel market and are under growing criticism for their failure in other industries (Callahan, 1987), (Kanet, 1987). One fundamental problem with current MRP-based systems is their independent handling of production scheduling and material planning. In so doing, MRP-based systems build in pre-planned lead times which must be large enough to accommodate all the potential variation encountered in the logistics process. What is needed are systems which simultaneously plan production and material.

OBJECTIVE

The concept of quick response through linkage has rightfully generated considerable discussion, but to date there has been little action in terms of the scientific design of techniques to implement it. Attempting to force-fit MRP-based planning methods to this problem appears ill-advised and clearly not consistent with the direction in technology towards networks of microcomputers.

Our objective in this project was to develop and evaluate several competing approaches to production planning which, in contrast to MRP-based approaches, simultaneously determine both a production schedule and material plan. We conducted the study by developing various scheduling algorithms and implementing them into a prototype interactive scheduling system.

APPROACH

The principal investigators for this project had already made considerable headway in advancing the theory of production planning. In Davis and Kanet (1991), the authors show how it is possible to design algorithms to efficiently account for economy of lot sizes and order scheduling in the same planning step. In the first phase of the project we developed an efficient algorithm which used the principles discussed in Davis and Kanet (1991) for solving production planning problems. We verified the correctness and speed of the new method by testing it on a carefully designed set of sample problems.

In the second phase of the project we incorporated the algorithm into a prototype production planning system suitable for demonstration to apparel manufacturers. The prototype system demonstrates how an apparel firm can quickly determine the impact of a change in the schedule, resulting from unanticipated events such as changes in customer requirements, breakdown of machines, or interruption in supply of input materials.

TECHNOLOGY TRANSFER

An important aspect of the project was the transfer of the knowledge gained so that others may build on it. To this end, we published the results of our evaluation at both research and apparel trade conferences, and demonstrated the algorithm at the Apparel Advanced Manufacturing Technology Demonstration (AAMTD) site. Our research results and the transfer thereof is particularly valuable to commercial and military software developers who supply manufacturing planning systems for apparel firms. Armed with the results of this project, these software developers are in a position to design software systems which provide real definition to the term "quick response" for the apparel industry.

BENEFITS

An operational planning system based on our prototype can improve the responsiveness of the entire apparel industry. Textile and apparel companies have realized the importance of working closely together and are establishing telecommunication networks to improve the interaction. Our focus here was on developing ways in which an individual firm can use this information effectively. Adopting an effective planning system will benefit individual companies (whether they produce raw materials, textiles, or apparel) and

therefore will improve the efficiency of the entire intra-industry network.

II. ALGORITHM DEVELOPMENT

INTRODUCTION

We address a class of scheduling problems in which there is a single machine with n jobs available for processing at time zero. Each job is identified by an integer from 1 through n, and each job i is described by these attributes:

p(i)	processing time,
s(i)	setup time for job i (possibly sequence dependent)

A schedule is defined as a vector of job completion times [C(1), C(2), ..., C(n)]. The objective is to find a schedule such that

$$g(C(1), C(2), \ldots, C(n))$$

is minimized, where g is a convex function in the usual mathematical sense; i.e., for any pair of schedules S = [C(1), C(2), ..., C(n)] and S' = [C'(1), C'(2), ..., C'(n)] and every a, 0 < a < 1,

$$g(aS + (1 - a)S') < ag(S) + (1 - a)g(S').$$

We refer to the objective function g as the "performance measure" and call the value of g the "cost" of a schedule. We assume that preemption is not allowed (Appendix A shows why this is not seriously restrictive). In the notation of Rinnooy Kan (1976), the problem may be described as n/1//convex, where "convex" indicates the type of performance measure.

BACKGROUND AND MOTIVATION

A standard assumption in the scheduling literature has been that performance measures are non-decreasing functions of job completion times (i.e., "regular" performance measures). The major scheduling monographs by Conway, Maxwell, and Miller (1967), Baker (1974), Coffman (1976), Rinnooy Kan (1976), and French (1982) all limit the scope of their presentation to scheduling problems with such regular performance measures. There exists a rich assortment of regular measures (e.g., mean weighted flow time, makespan, mean weighted tardiness, etc.) so that in many scheduling situations the assumption of regularity is not a major limitation. But there are many important occasions when non-regular measures apply. One such class of problems includes those in which it is important to minimize some measure of variation in flowtime

or waiting time among the jobs to be serviced. These types of problems have been studied by Merten and Muller (1972), Schrage (1975), Eilon and Chowdhury (1977), Kanet (1981b), and Vani and Raghavachari (1987). Recently, Bagchi (1987) reported on a class of non-regular problems in which the objective was to minimize a weighted sum of mean and variance of flowtimes or waiting times.

Clearly, regular measures comprise a proper subset of convex performance measures. Of primary concern here are problems in n/1//convex which have non-regular performance measures. A notable illustration of this type of problem occurs when jobs incur penalties for both earliness and tardiness. Consider for example the case when g, the performance measure, is the sum over all jobs of

$$e(i)\max\{d(i)-C(i),0\} + t(i)\max\{C(i)-d(i),0\}, \qquad (1)$$

where d(i) is job i's due date, and e(i) and t(i) are early and tardy penalty coefficients, respectively.

We call this particular problem n/1//ET, or "ET" for short. The ET problem is NP-hard, since it is a generalization of the weighted tardiness problem, which is known to be NP-hard (Lenstra, Rinnooy Kan, Brucker 1977). Few research studies about ET have been reported. Sidney (1977) studied a version of ET in which the objective was to minimize the maximum penalty subject to certain restrictions on the due dates of the jobs. Kanet (1981a), Sundararaghavan and Ahmed (1984), Hall (1986), and Bagchi, Sullivan, and Chang (1986) reported on the case when all e(i)=t(i) and there is a common due date for all jobs. Ow and Morton (1988, 1989) investigated using a heuristic search to solve the more general version of problem ET, but they made the restrictive assumption that the machine is operated in a non-delay mode so that schedules with embedded idle time were not permitted. Faaland and Schmitt (1987) addressed the general ET problem under the complication of multi-machines and product structures. They approached the problem in two phases by first heuristically assigning the sequence of operations at each work center so that the product structure constraints were satisfied. In phase two they solved the idle time assignment problem at each resource as an independent maximum flow problem. Fry, Armstrong and Blackstone (1987) also provided a procedure for optimally inserting idle time into a given sequence, formulating the problem as a linear program. A comprehensive review of the literature on ET has been recently provided by Baker and Scudder (1988).

Problem ET is important to study because there are many applications (especially in industry) where penalties are

incurred for both early and tardy job completion. As Kanet and Christy (1984) claim, a typical constraint in industrial settings is the forbidding of early shipment to customers. When this is the case, early job completion can cause the cash commitment to resources in a time frame earlier than needed, giving rise to early completion penalties. Tardiness penalties arise from a variety of sources -- loss of customer good will, opportunity costs of lost sales, and direct cash penalties. The ET problem is consistent with the current focus in industry toward Just-in-Time production.

Figure 1 illustrates several possible variations of the ET problem. In (1.a) the penalties for each job are the same whether the job is early or tardy, rendering "absolute lateness" as the criterion. (As is usual in the scheduling literature, we define lateness for job i as C(i) - d(i)). In (1.b) the penalty for a given job is pictured to describe the functional form of Expression (1). Figure (1.c) depicts a non-linear case in which penalties may grow at an increasing rate as a job's completion time deviates from its due date. An example of this situation would be a case in which penalties arise from cash flows so that the effect of "compounding" comes into play.

Organization

This part of the report extends previous results by generalizing to the larger class of convex performance measures. We describe and demonstrate a polynomial algorithm (TIMETABLER) for finding the optimum assignment of idle time for a given permutation of jobs.

In the next section we show that the solution domain for single machine scheduling problems is comprised of disjoint convex sets, and show how this leads to a generalization of the notion of a dominant set of semi-active schedules. We then provide some important definitions and clarifications and describe a polynomial time procedure for optimally inserting idle time into a given permutation of jobs. We then show how the procedure can be employed within a branch-and-bound algorithm, and report our initial computational experience. Finally, we present additional computational results of an investigation of the influence of problem parameters, and summarize our major conclusions.

SOME CLARIFICATIONS

Convexity of the Performance Measure

Generally, the notion of a convex function applies to situations in

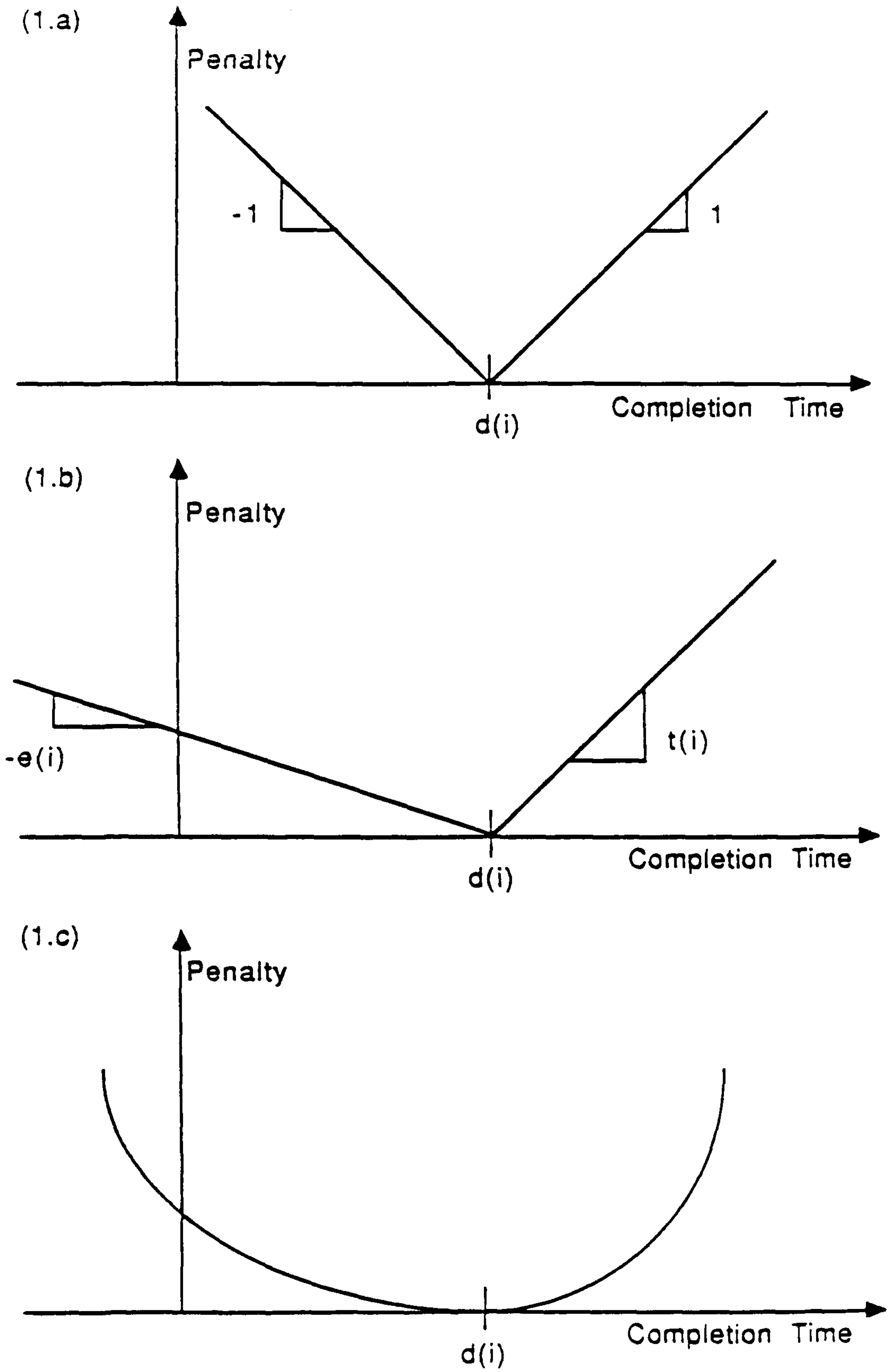


Figure 1. Examples of Different Job Penalty Functions for the ET Problem

which the domain of the function is itself a convex set. This is not the case with the scheduling problems which we address. This issue arises as a direct consequence of the implicit constraint that the machine cannot process more than one job at a time. To illustrate this, consider a simple two-job scheduling problem such that $p(1)=1$ and $p(2)=4$. The shaded areas of Figure 2 identify the feasible solutions to this problem, clearly illustrating that the feasible solution set is not convex. Unshaded areas correspond to solutions which require the machine to process more than one job at a time. Note how the two distinct convex regions are identified with one of the possible permutations of the two jobs. As shall be presently proven, this will generally be the case; i.e., all solutions corresponding to a particular permutation form a convex set. First some definitions are needed.

Definition. $S(P)$ is said to be a schedule S defined over the permutation P such that $S(P) = [C(1), C(2), \ldots, C(n)]$. To simplify the notation we assume that $C(1) < C(2) < \ldots < C(n)$ and that the job indices $1,2,\ldots,n$ are in the order specified by P. $S(P)$ is said to be feasible if $C(i) > C(i-1)+p(i)+s(i)$ for $i = 1,\ldots,n$ where the value of $C(0)$ is defined as 0.

Definition. A set A, of schedules, is said to be convex if for every S, S' in A, the schedule S'' defined by

$$S'' = a(S) + (1 - a)S' \text{ , where } 0 < a < 1,$$

is also in A.

Theorem 1. Given any permutation P, the set $F\{P\}$ of all feasible schedules defined over P is a convex set.

Proof: Assume the theorem is false, namely assume that there exist two schedules $S(P) = [C(1), C(2),\ldots, C(n)]$ and $S'(P) = [C'(1),C'(2),\ldots, C'(n)]$, such that

$S'' = [aC(1)+(1-a)C'(1),aC(2)+(1-a)C'(2),\ldots,aC(n)+(1-a)C'(n)]$

for some a, $0 < a < 1$, is not in $F\{P\}$. Then for at least one job j, it must be that

$aC(j)+(1-a)C'(j) < aC(j-1)+(1-a)C(j-1)+p(j)+s(j)$, or

$$a[C(j) - C(j-1)] + (1-a)[C'(j)-C(j-1)] < p(j)+s(j), \qquad (2)$$

otherwise S'' would be in the feasible set $F\{P\}$ contrary to the assumption. Since both S and S' are feasible it follows that

$$a[C(j) - C(j-1)] > a(p(j)+s(j)), \qquad (3)$$

and $(1-a)[C'(j) - C(j-1)] > (1-a)(p(j)+s(j))$.

(4) Adding (3) and (4) yields

$a[C(j)-C(j-1)] + (1-a)[C'(j)-C'(j-1)] > p(j)+s(j)$

which directly contradicts (2). Q.E.D.

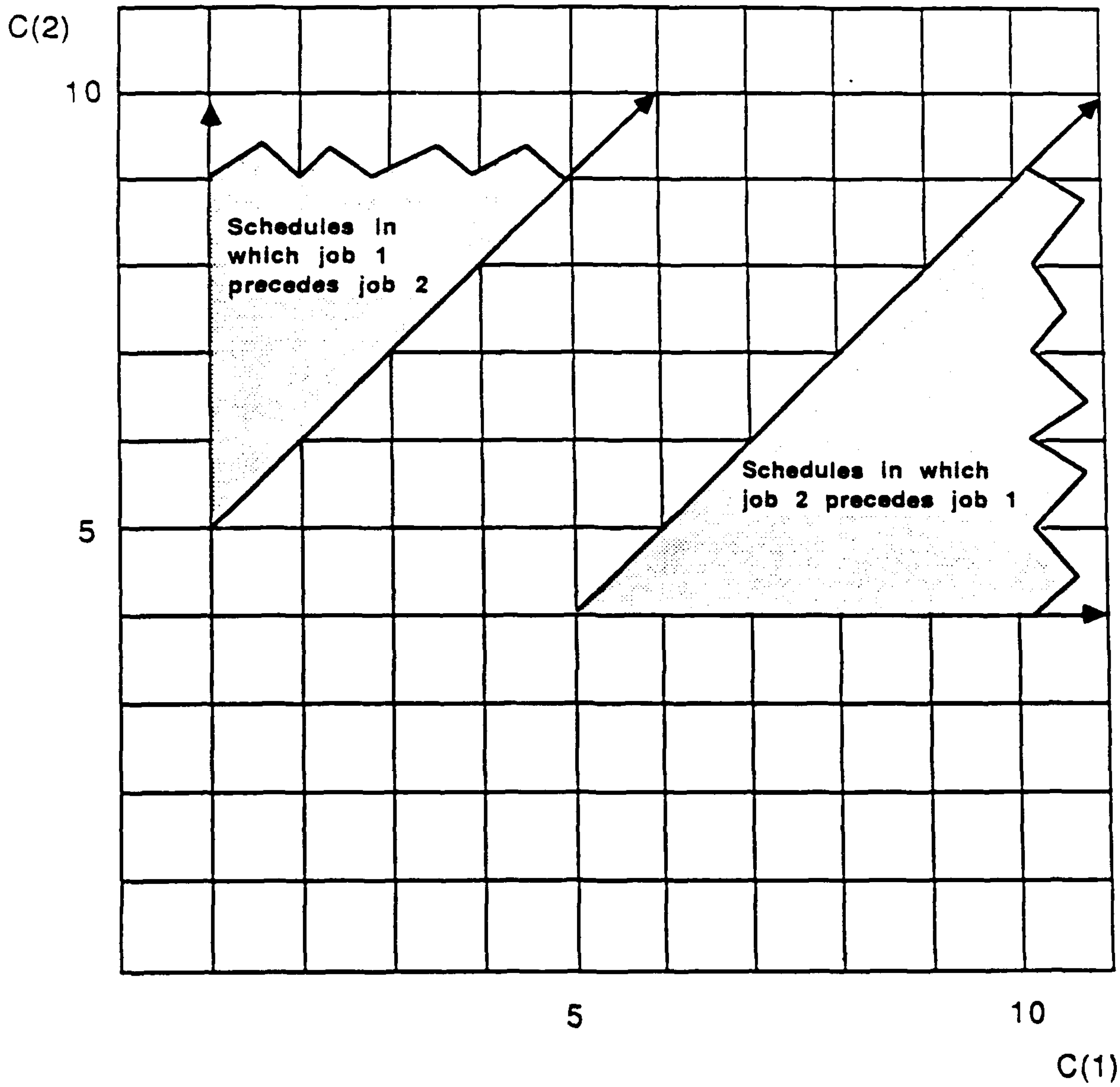


Figure 2. Feasible Solution Space for a Two-Job Problem with P(1) = 1 and P(2) = 4

Semi-Active Schedules

In this section we extend the notion of semi-active schedules, defined by Giffler and Thompson (1960) and Baker (1974), to n/1//convex. In the terminology used by Baker (1974), a semi-active schedule is one in which no "local left shift" can be made. A local left shift is accomplished by decreasing the start time of some job while preserving the job sequence. A similar notion of semi-active also applies to problems with convex measures.

Because an arbitrary amount of idle time may be inserted between adjacent jobs in a schedule, an infinite number of feasible schedules may exist for a particular scheduling problem. In the case of a regular measure, it turns out that in searching for an optimum schedule we need only consider the finite set of semi-active schedules. In other words, the semi-active schedules dominate the set of all schedules. An enumerative search for an optimum solution may be organized in terms of job permutations, since there exists a polynomial time procedure for converting a permutation to the associated semi-active schedule. (The procedure is trivial: Schedule the jobs in the desired sequence so that the machine is never idle prior to the start of a job.)

In adapting Baker's terminology to n/1//convex, we call "semi-active" those schedules whose cost cannot be reduced by "local shifting" a job. "Local shifting" is defined as altering the completion time of a job without changing the job sequence. Local shifting thus includes both "local left shift" and "local <u>right</u> shift" in Baker's terms. Moreover, local shifting a job may displace the completion times of entire groups of jobs.

The simple examples in Figure 3 are instances of the ET problem, described earlier, with $e(1) > t(2)$. The examples illustrate the conversion of a non-semi-active schedule to a semi-active schedule. That semi-active schedules dominate the set of all schedules follows directly from the definition.

GENERATING SEMI-ACTIVE SCHEDULES

Procedure for Generating a Semi-Active Schedule

In this section we describe procedure TIMETABLER, which converts a job permutation to a semi-active schedule in polynomial time. The following notation is adapted from that used by French (1982).

t	the stage number.
S(t,P)	a partial schedule of (t - 1) jobs, where P is the permutation of jobs.

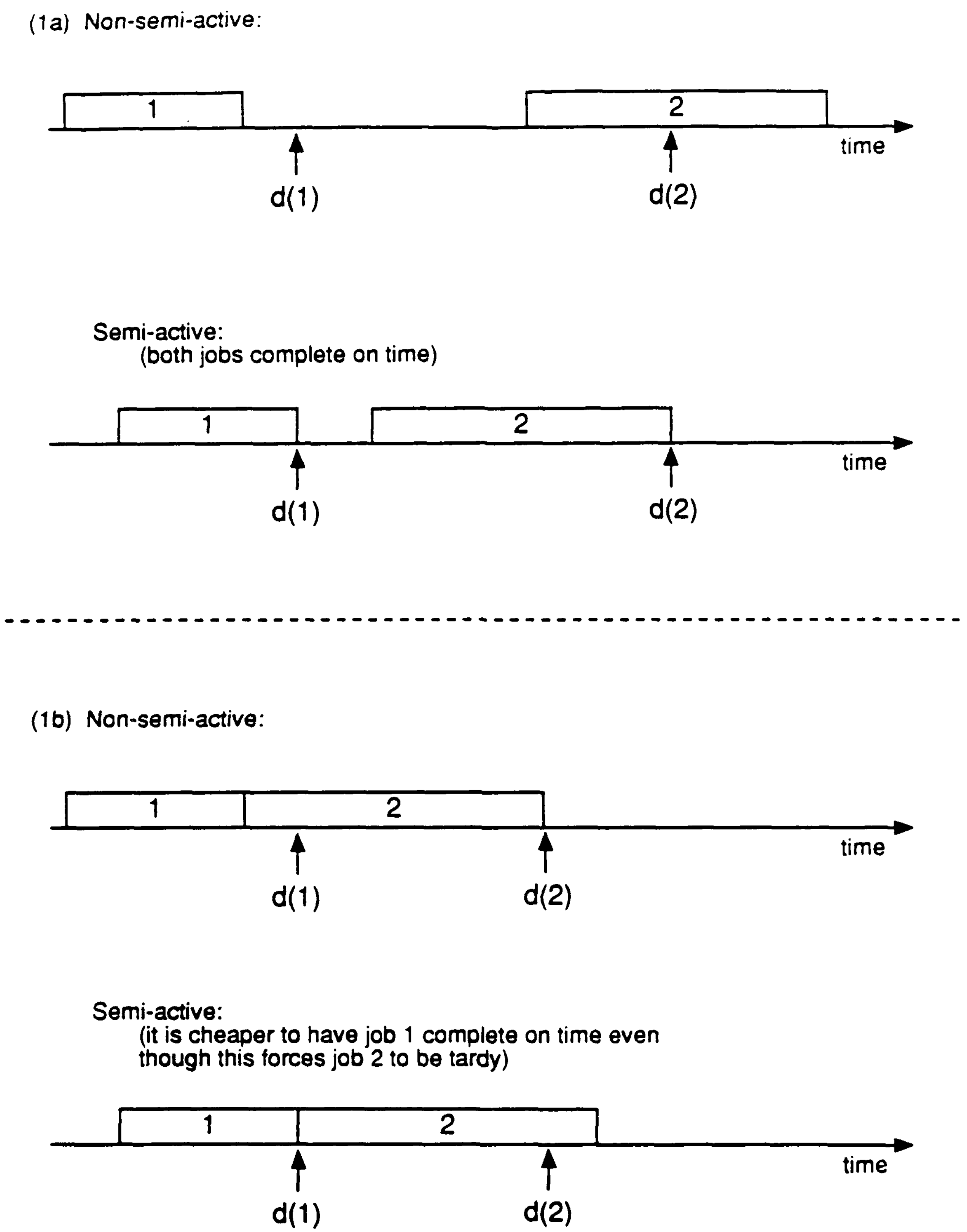


Figure 3. Examples of semi-active schedules, where $e(1) > t(2) > 0$

We treat P as an ordered set and let jP denote the permutation in which P is immediately preceded by job j.

P' — an ordered set of unscheduled jobs; jobs that will follow those in P' are in P; i.e., P'P is the complete job permutation at any stage.

<u>Procedure</u> <u>TIMETABLER</u>:

Step 1 Let t = 1 with S(t,P) being null and P being empty.
Step 2 Select the last job j in P', and delete it from P'.
Step 3 Move to the next stage by
- 3.1 adding j to S(t,P) to create S(t+1,jP) such that increasing C(j) would not reduce cost of S(t+1,jP), using the following steps:
  - 3.1.1 Initially job j is started at time 0, pushing jobs in S(t,P) to the right (so as to make them later) if necessary.
  - 3.1.2 Shift job j to the right until the marginal cost of so doing is not negative.
- 3.2 incrementing t by 1;
- 3.3 setting P = jP.

Step 4 If P' is not empty, go to Step 2. Otherwise, stop.

<u>Theorem</u> <u>2</u>. TIMETABLER produces a semi-active schedule.
<u>Proof</u>. The proof is by induction on the stage, t. For t = 1, the job selected in Step 2 will obviously be scheduled in Step 3 such that no shifting could reduce cost. Assume that for q, 1 < q < n, that S(q,P) is semi-active. Let k(q) designate the job selected in Step 2 at stage q+1. Then assume that S(q+1,k(q)P) is not semi-active, meaning that there must be some job u which can be shifted so as to reduce the cost of S(q+1,k(q)P). We shall show that this assumption leads to a contradiction, by proving that any such shift cannot reduce cost. We may identify certain groups of jobs in S(q+1,k(q)P) as in Figure 4.

Any of the named groups of jobs, excluding k(q), could be null, and idle time could be at any of the boundaries.

SHIFTED contains the jobs which have been displaced (pushed to the right) by the positioning of k(q).

UNSHIFTED contains the jobs which have not been moved by the positioning of k(q). Thus, P is the union of the jobs in SHIFTED and UNSHIFTED.

<u>Case</u> <u>1</u>: u is in UNSHIFTED. No movement of u can reduce the cost contribution of the UNSHIFTED jobs; otherwise there is a contradiction of the assumption that S(q,P) is semi-active.

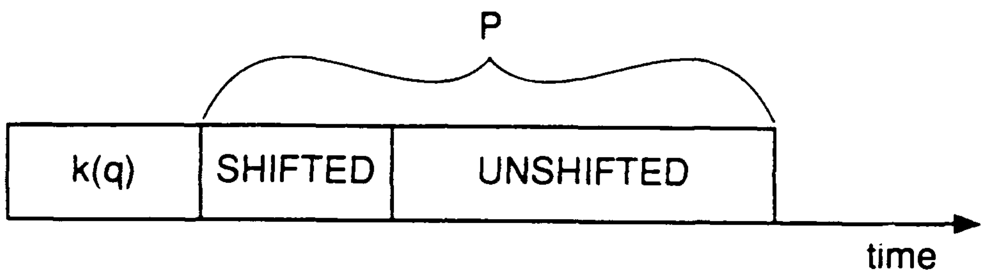


Figure 4. A schedule at stage q + 1 of TIMETABLER

Case 2: u is in SHIFTED. Any shift to the right cannot reduce cost of S(q+1,k(q)P), else it would also have reduced cost of S(q,P), a contradiction that S(q,P) was semi-active. Any shift to the left would displace all jobs from k(q) through u in the schedule. This shift cannot reduce cost of S(q+1,k(q)P) because of the function of Step 3.1 in the previous stage.
Case 3: u is k(q). Shifting u to the right cannot reduce cost, otherwise this would have already been done in Step 3.1 of the previous stage. Shifting u to the left cannot reduce cost, otherwise u would not have been placed in its current position by Step 3.1 of the previous stage.

Since no shift of a job u can decrease cost, S(q+1,k(q)P) is semi-active. Q.E.D.

TIMETABLER is a polynomial time procedure if g is computable in polynomial time. To see this, note that the main loop requires O(n) executions (i.e., there are O(n) stages). Step 3.1.1 is O(n). Step 3.1.2 is a loop, such that g is re-evaluated at a constant number of discrete time intervals in accordance with the desired precision of the solution. Thus the computation in the entire Step 3.1 is O(n). For TIMETABLER as a whole, the computational effort is O(n**2).

Relation to Economic Batch Scheduling

Theorem 3. If g is computable in polynomial time, TIMETABLER produces a semi-active schedule in polynomial time, even if jobs have sequence-dependent setup times.
Proof: In the case of sequence-dependent setup times, the setup time may be included as part of the processing time of job j in Step 3.1 of TIMETABLER. The procedure then produces a semi-active schedule as before. Accounting for sequence-dependent setup time contributes only a constant amount of computation, since it may be accomplished by a simple reference to the setup time matrix followed by an addition of setup time to the processing time. Thus TIMETABLER is still a polynomial time procedure. Q.E.D.

Theorem 3 is important because it (along with the assumption of no preemption) makes it possible to interpret our results beyond problems of pure sequencing. For example, consider the problem of simultaneously determining the batch sizes and sequence when a single machine is confronted with a set of orders for different products. Each order is characterized by a product type, a required quantity, and a due date. Changeover times (and costs) to go from product type i to type j are known. This is the classic economic batch scheduling problem (EBS) with discrete deterministic demand. Clearly, EBS is a special case of n/1//convex. For

any instance of EBS we simply "prepare" the data by breaking each order into "atomic" jobs (i.e., jobs which for practical purposes cannot be split) and creating a setup time matrix which accounts for job changeover. If job j and job i are of the same type then their changeover time is zero; otherwise the changeover time is taken from the product type changeover times of the original problem statement.

Numerical Example Illustrating TIMETABLER

The following numerical example illustrates TIMETABLER. Consider the seven-job sample data in Table I. In addition to the processing time and due date information, the table shows the earliness and tardiness coefficients, e and t respectively. We assume that each job has a cost function of the form given by Expression (1) so that the example problem is an instance of n/1//ET. Figure 5 shows how TIMETABLER would develop a schedule (a timetable) under the assumption that the jobs in Table I are to be arranged in the sequence 7, 6, 5, 4, 3, 2, 1, the initial setting of P'.

In Figure 5 we see that in step 2 job 1 is added to the null schedule so that increasing $C(1)$ does not reduce cost. This occurs by setting $C(1) = d(1)$. At stage 3 job 2 is added before 1 and $C(2)$ is set to $d(2)$ since $e(2) > t(1)$, causing $C(1)$ to shift from 30 to 33. In stage 4 job 3 is added with $C(3) = d(3)$, which does not affect jobs 1 or 2. At stage 5, job 4's completion time is increased until it would affect $C(3)$ because $e(4) < t(3)$. In stage 6 job 5 shifts $C(4)$ and $C(3)$ but not $C(2)$ or $C(1)$ because $t(3) < e(5) + e(4) < t(3) + t(2) + t(1)$. In stage 7 job 6 is added without affecting any other job completion times. Finally, in stage 8 job 7 is given the minimum possible completion time.

USING TIMETABLER IN AN ENUMERATIVE SEARCH

Procedure TIMETABLER may enhance an enumerative search in two ways:

1. TIMETABLER efficiently produces a semi-active schedule from a job permutation, thus permitting a search over the finite domain of permutations.
2. The cost of a partial semi-active schedule produced by TIMETABLER can be used as a lower bound on the cost of any full schedule extended from it.

We illustrate the usefulness of TIMETABLER in a branch-and-bound algorithm, which operates by splitting the set of solutions into subsets. Each partial schedule, $S(t,P)$, represents the subset of solutions such that P, an ordered set of jobs, constitutes the last (t-1) jobs in each solution. TIMETABLER generates the partial schedules under

Table I
Example Scheduling Problem

Job Number	Processing Time (p)	Due Date (d)	Early Penalty (e)	Tardy Penalty (t)
1	8	30	11	15
2	2	25	16	18
3	6	16	10	12
4	3	18	9	12
5	7	26	18	38
6	2	4	20	25
7	3	12	10	12

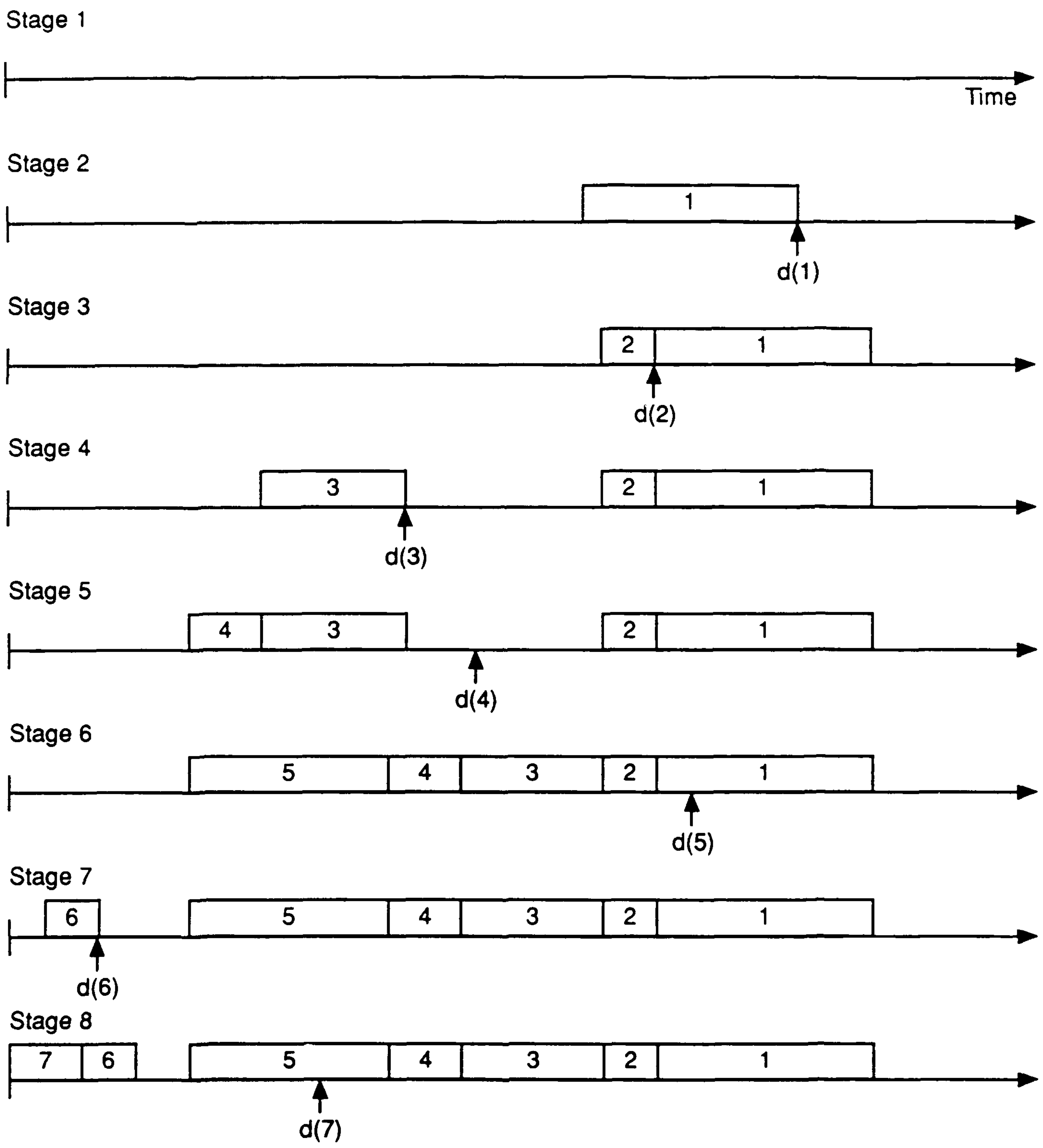


Figure 5. Application of procedure TIMETABLER to a seven-job sample problem

the guidance of the branch-and-bound algorithm, which chooses the job to be put last, next to last, etc.

The cost of the partial schedule S(t,P) is a lower bound (optimistic estimate) on the cost of any full schedule generated from it (a proof is in Appendix B). The lower bound could be increased (and thus improved in quality) by constraining the addition of j to S(t,P) in step 3 of TIMETABLER, such that j starts no earlier than the sum of the s(i) and p(i) for i in P. (We could improve this bound slightly by adding to it the minimum setup time for all i in P, given that i follows j.)

We assume that the branch-and-bound method uses the lower bound in the traditional way. We may eliminate any partial schedule (subset of solutions) which has a lower bound greater than the cost of the lowest cost full schedule which has been produced.

Preliminary Computational Results

We performed some computational work in order to get a preliminary indication of the usefulness of TIMETABLER in reducing schedule cost, and to determine run times for a set of sample problems. The sample problems were adapted from the 16 different 8-job problems found in Baker (1974, p.289). The processing times and the due dates of the jobs were left as originally specified. However, instead of weighted tardiness the cost of a schedule was mean absolute lateness (MAL) defined as

$$\text{MAL} = \text{sigma}_{i=1}^{n} |C(i)-d(i)|/n$$

which is well-known in the scheduling literature. MAL is convex and non-regular, making it possible that an optimum solution might contain embedded idle time.

TIMETABLER was employed two ways. First, it was used in the fashion outlined in the preceding paragraphs -- i.e., by employing it at each node in the search tree. This procedure guaranteed finding an optimum solution by explicitly considering embedded idle time at each stage of the search. A second approach was to find the optimum solution under the constraint that no idle time be permitted prior to any job. (This has been a common assumption in the literature; e.g., see Ow and Morton, 1988.) TIMETABLER was then employed once to optimally allocate idle time to the solution to this constrained problem.

The algorithms were implemented in C under BSD UNIX 4.2, running on a Celerity C-1200 minicomputer. Table II summarizes the results. The elements of the second column of the table represent the objective function value for the optimum solution (Case 1). The third column indicates the percent increase in the objective function of the optimum solution to the "no idle time permitted" problem (Case 2). The fourth column in the table shows the percent increase in the objective (from optimum) when TIMETABLER is applied to the Case 2 schedule forming the timetabled Case 2T schedule.

The computer processor time required to find Case 1 schedules varied from .3 to 157.5 seconds with an average time of 29 seconds for the 16 problems. The times to find Case 2 schedules varied from .1 to 14.3 seconds, with an average time of 7 seconds. As can be seen, in many of the cases there was no change in cost between Case 1 and Case 2 as both procedures yielded the identical schedule. This occurred because the due dates to these problems were clustered relatively close together near time zero, causing the optimum solution to the unconstrained problem to have no embedded idle time.

The result for Problem 9 is a notable exception. Figure 6 provides a Gantt chart for Cases 1, 2 and 2T associated with Problem 9. Comparing Case 1 with Case 2 allows us to see how the simplifying assumption of no idle time can lead to a significantly inferior schedule. The cost for Case 2 was 723, 39% higher than the optimum value of 520 yielded by Case 1. Case 2T illustrates the mitigating effect of applying TIMETABLER once, after a final permutation has been determined (Case 2's performance is improved to 631). However, it also shows that such one-shot applications of TIMETABLER may still provide schedules which are non-optimum. This example is particularly interesting because Case 2T and Case 1 both had an idle period only before the start of the first job, yet the cost for Case 2T was still 21% worse than for Case 1. The intriguing point here is that good schedules may not necessarily be found simply by concentrating first on finding a good permutation and then "sprinkling in" the appropriate amount of idle time to form a schedule. As this example illustrates, the explicit consideration of idle time during the search can meaningfully change the resulting permutation.

FURTHER COMPUTATIONAL RESULTS

A closer examination of the Baker problems was conducted to get a clearer understanding of the factors which might influence the results. Except for Problem 9, the optimum solution to each problem contained no idle time. One factor which might affect the amount of idle time in an optimum

Table II
Results for Baker's (1974) Problems
When the Cost is MAL (e=1, t=1, for each job)

	COST	% ABOVE OPTIMUM	
Baker Problem	Case 1	Case 2	Case 2T
1	956	0	0
2	1092	0	0
3	480	0	0
4	1043	0	0
5	868	3.10	3.10
6	975	0.10	0.10
7	934	0	0
8	1641	0	0
9	520	39.40	21.30
10	1735	0	0
11	695	0	0
12	1095	1.20	1.20
13	599	0	0
14	1328	0	0
15	708	0	0
16	1659	0	0

Case 1: Optimum solution
Case 2: Optimum solution when idle time is not allowed
Case 2T: Case 2 solution after applying TIMETABLER

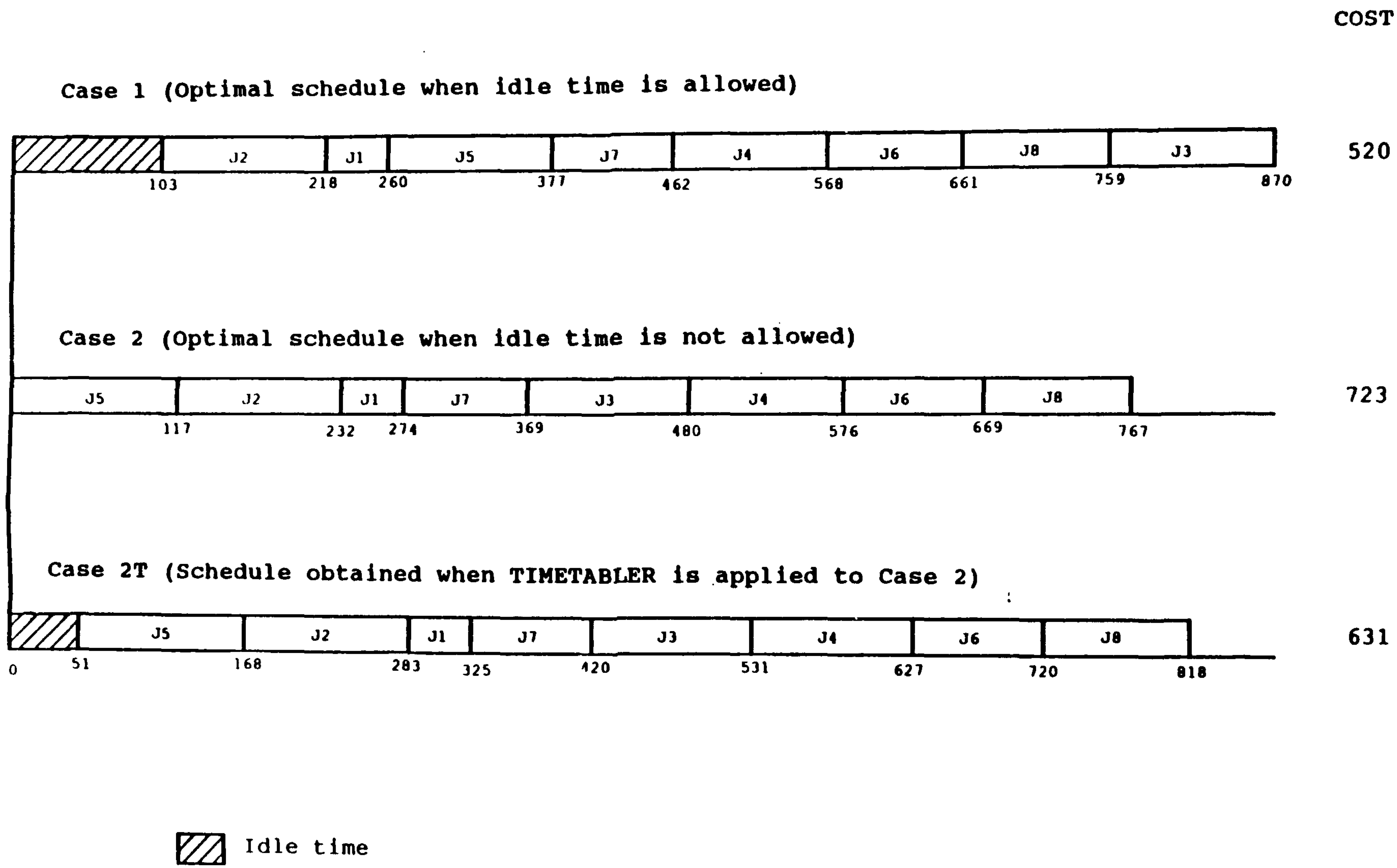


Figure 6. Applying Procedure TIMETABLER to Baker's (1974) Problem 9

schedule is the aggregate level of due dates. In constructing sample problems, Srinivasan (1971) and Baker and Martin (1974) have used a parameter called the "tardiness factor", Tau, defined as:

Tau = 1-(dbar/n(pbar)),

where dbar and pbar stand for mean due date and mean processing time, respectively. Tau reflects the aggregate level of due dates in the form of a crude prediction of the percent of the jobs that will be tardy. For the 16 Baker problems, the average Tau value was 0.51. For his studies, Baker was interested in comparing different algorithms for minimizing tardiness. A relatively large value of Tau was thus sensible for reasons of making sure differences in algorithm performance could be detected. Too small a value of Tau might result in many occurrences in which each algorithm found a solution with zero tardiness, so it was prudent for Baker's purposes to have such high values. However, practical scheduling situations would likely have Tau values lower than 0.51. In a recent study of the ET problem, Ow and Morton (1989) generated two sets of sample problems with Tau values set at 0.2 and 0.6, respectively(*).

With this background, we continued our examination of the Baker problems by modifying them in the following way. For each problem instance we computed the Tau value. We then added a constant to each job's due date so that every problem instance had a Tau of 0.2. This provided us with problems with substantially lower Tau values than those originally constructed by Baker, yet still comparable to those constructed by Ow and Morton. We then repeated the procedure described in the previous section.

The Influence of Tau

Table III shows the results for these modified problems in the same format as Table II. Tau appears to have a major influence on the results. As the second column of the table shows, ignoring idle time during the search always led to sub-optimum results. Except for Problem 16, applying TIMETABLER to the no-idle-time-permitted optimum schedule made a substantive improvement; but in every case this "one-shot" application of TIMETABLER delivered a schedule that was still far from optimum.

The Influence of the Due Date Range Factor R

Aside from the parameter Tau, another factor that might influence the results is the degree to which due dates are spread out. Let dmax (dmin) represent the maximum (minimum) due date among the jobs to be scheduled. Baker and Martin

Table III
Results for Baker's (1974) Problems Modified With Tau=0.2
When the Cost is MAL (e=1, t=1, for each job)

	COST	% ABOVE OPTIMUM	
Baker Problem	Case 1	Case 2	Case 2T
1	636	102.67	24.69
2	1042	33.01	19.77
3	336	272.02	180.95
4	974	29.47	21.87
5	868	86.87	44.24
6	964	37.03	22.30
7	768	66.41	27.60
8	1485	23.16	6.33
9	520	164.62	78.08
10	1545	34.43	24.08
11	215	425.12	268.84
12	1095	24.66	11.51
13	549	181.60	131.33
14	1287	34.34	13.68
15	341	308.21	238.42
16	1511	29.19	29.19

Case 1: Optimum solution

Case 2: Optimum solution when idle time is not allowed
Case 2T: Case 2 solution after applying TIMETABLER

(1971) and Ow and Morton (1989) used the so-called "due date range factor", R defined as

$$R = (dmax-dmin)/n(pbar)$$

as a measure of due date variability. We calculated R for each of the Baker problems and sorted them in order of non-decreasing R. The R values for Baker's problems ranged from 0.087 to 0.778 and were comparable to those studied by Ow and Morton where problem instances were created by drawing R from a uniform distribution with mean of 0.4 or 1.0. (Note that Tau and R are independent so that our modifications of the Tau values to Baker's problems left the original R values intact.) Table IV summarizes some of our additional results. Note that the effect of R is not perfectly controlled since the problem instances associated with each R value are in fact different. Nevertheless, the influence of R on the results appears to be very strong. At least in the range of R values for these problems, higher R generally meant greater deviation from optimum for each of the different scheduling procedures. (A major exception is case 3T which is discussed below.)

Note that the columns of Table IV have been arranged in pairs {Case 2, Case 2T}, {Case 3, Case 3T}, and so on, such that the second member of each pair represents the sequence found after applying TIMETABLER to the first pair member -- i.e., after inserting idle time. Observing these pairings allows us to see the relative improvement that application of TIMETABLER provides.

Earliest Due Date Versus Search Algorithms

Case 3 of Table IV corresponds to the simple heuristic of scheduling jobs according to earliest due date (EDD). Case 3T is the EDD schedule with idle time inserted according to TIMETABLER. What is rather surprising is the quality and robustness of this simple heuristic. Compare the results in the columns labelled Case 3 and 3T to their counterparts in the columns Case 2 and 2T. As expected, Case 3 results are dominated by Case 2 (Case 2 is optimum when idle time is forbidden). Yet comparison of Case 3T with 2T leads to something dramatically different. Except in one instance (Problem 2), the use of the simple EDD rule combined with the application of TIMETABLER led to a better solution than the one found through branch-and-bound applied to the "no idle time permitted" problem -- even after the resulting sequence was timetabled by TIMETABLER. Moreover the quality of the EDD timetabled schedules (unlike the other procedures) was largely insensitive to the range factor R. These results are rather surprising not only from the point of view that EDD is so simple -- it is O(nlog(n)) -- but also because it uses no information about the form of the objective function.

Table IV
Additional Computational Results
for Baker's Problems Modified Such That Tau=0.2
When the Cost Function is MAL (e=1, t=1, for each job)

Baker Problem	Range Factor	COST	% ABOVE OPTIMUM					
		Case 1	Case 2	Case 2T	Case 3	Case 3T	Case 4	Case 4T
4	0.087	974	29.47	21.87	53.49	4.41	82.14	82.14
12	0.101	1095	24.66	11.51	51.96	10.14	42.74	42.74
10	0.111	1545	34.43	24.08	53.59	11.00	42.39	42.39
8	0.116	1485	23.16	6.33	37.04	5.12	35.96	35.96
14	0.166	1287	34.34	13.68	44.13	5.44	35.82	35.82
16	0.167	1511	29.19	29.19	42.69	3.64	44.41	44.41
2	0.197	1042	33.01	19.77	64.78	19.87	60.08	60.08
6	0.207	964	37.03	22.30	69.19	9.02	76.66	76.66
5	0.412	868	86.87	44.24	105.65	6.57	107.60	90.32
7	0.452	768	66.41	27.60	82.81	20.57	82.55	82.55
1	0.543	636	102.67	24.69	134.28	19.18	116.19	66.19
9	0.638	520	164.62	78.08	207.12	17.12	200.96	100.38
13	0.653	549	181.60	131.33	239.71	21.13	210.75	96.36
3	0.667	336	272.02	180.95	325.60	0.00	287.80	245.54
11	0.696	215	425.12	268.84	534.88	0.00	529.30	514.42
15	0.778	341	308.21	238.42	351.61	0.00	323.46	247.51

Case 1: Optimum solution

Case 2: Optimum solution when idle time is not allowed
Case 2T: Case 2 solution after applying TIMETABLER

Case 3: Earliest Due Date (EDD) with no idle time
Case 3T: Case 3 solution after applying TIMETABLER

Case 4: Minimum tardiness solution
Case 4T: Case 4 solution after applying TIMETABLER

For purposes of comparison, we found a solution (through branch-and-bound) to each problem which minimized total tardiness. The entries in the columns labelled Case 4 and 4T show the quality of these solutions when the objective is MAL. Case 4 can be thought of as the case when earliness costs are ignored.

Sequence versus Idle Time

Closer examination of the results revealed that except for Problems 1 and 15, the optimum schedules contained no idle time other than prior to the start of the first job in the sequence. This rather intriguing result led us to the following observation: what differentiated optimum solutions from others was not so much the existence of idle time but the fact that the sequences themselves were different. This point is illustrated in Figure 7 which shows the Gantt charts for all the cases for the modified Problem 9. In the figure, the due date of each job is shown by a vertical line situated behind the block diagrams. Tardy jobs are indicated by cross-hatching of the jobs' associated blocks; jobs completed on their due dates are indicated by shaded blocks; start times for contiguous blocks of jobs are shown immediately above the blocks. As the figure shows, the optimum schedule (Case 1) differs from the others not only because of the amount of idle time but also because the sequences are very different. For example, the Case 1 schedule (the optimum schedule) and the Case 2T schedule had one idle period occurring before the start of the first job in the sequence. But the sequence for Case 1, {J2,J1,J5,J7,J4,J6,J8,J3} differed markedly from that of Case 2, {J5,J2,J3,J1,J7,J4,J6,J8}. Why this occurs can be explained directly in terms of the deployment of TIMETABLER. For Case 1, TIMETABLER was deployed within the branch-and- bound search each time a lower bound was calculated for a partial schedule. For Case 2, an erroneous lower bound was calculated since idle time was never inserted into a partial schedule. This resulted in overstatement of the cost of a partial schedule and the erroneous discarding of nodes during the search procedure. As our results show, this can yield far from optimum solutions.

Idle Time Allocation

As R is increases one would expect the number of occurrences of idle periods within the optimum schedule to increase. Our results confirmed this expectation. This is illustrated in Figure 8, where the Gantt charts for Problem 15 (the problem with the largest R) are depicted. Figure 8 shows that the optimum schedule (Case 1) for the modified Problem 15 had three idle periods -- one before the start of the first job, the other two dispersed within the sequence. Case 2T had two idle periods, the second period occurring in the time interval (488,493). The timetabled EDD schedule (Case 3T) was

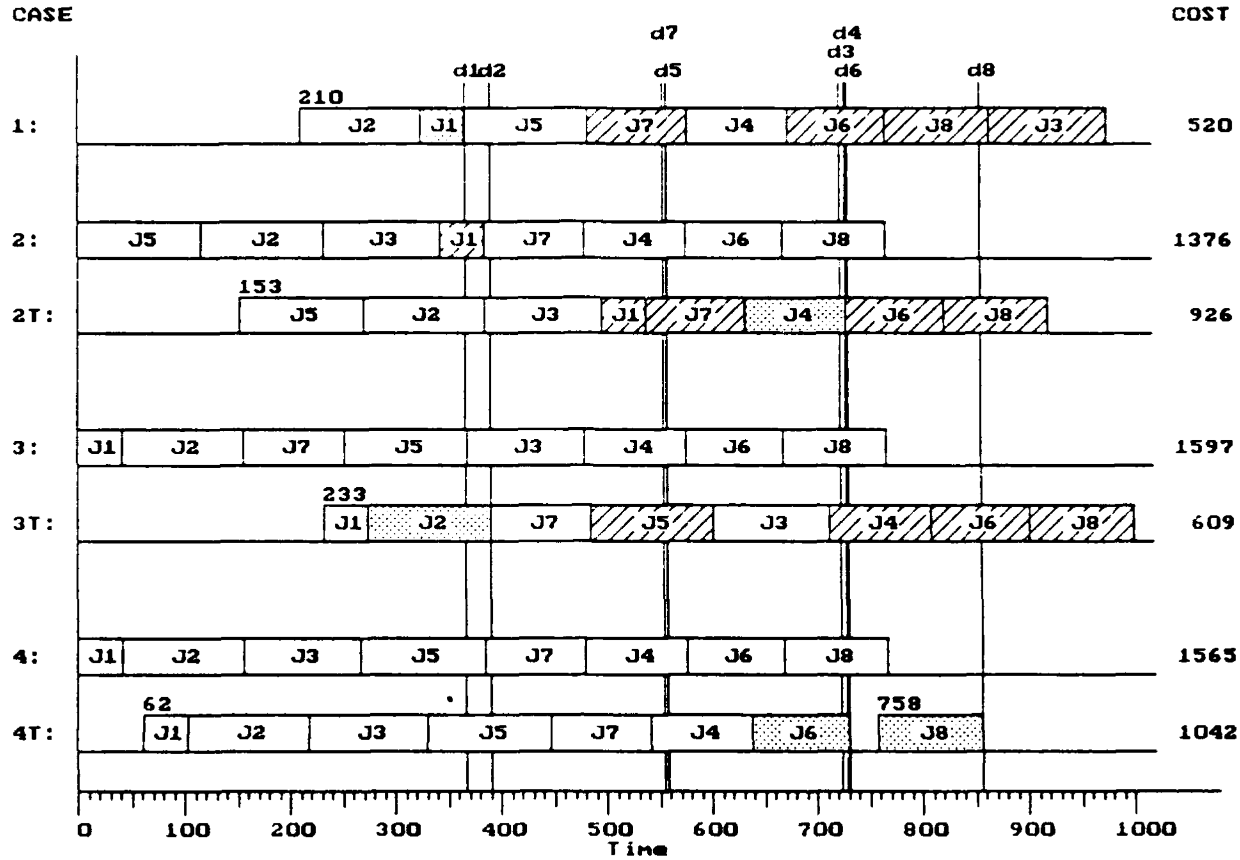


Figure 7. Gantt Charts for the Solutions to Baker's Modified Problem 9

Case 1: Optimum solution

Case 2: Optimum solution when idle time is not allowed
Case 2T: Case 2 solution after applying TIMETABLER

Case 3: Earliest Due Date (EDD) with no idle time
Case 3T: Case 3 solution after applying TIMETABLER

Case 4: Minimum tardiness solution
Case 4T: Case 4 solution after applying TIMETABLER

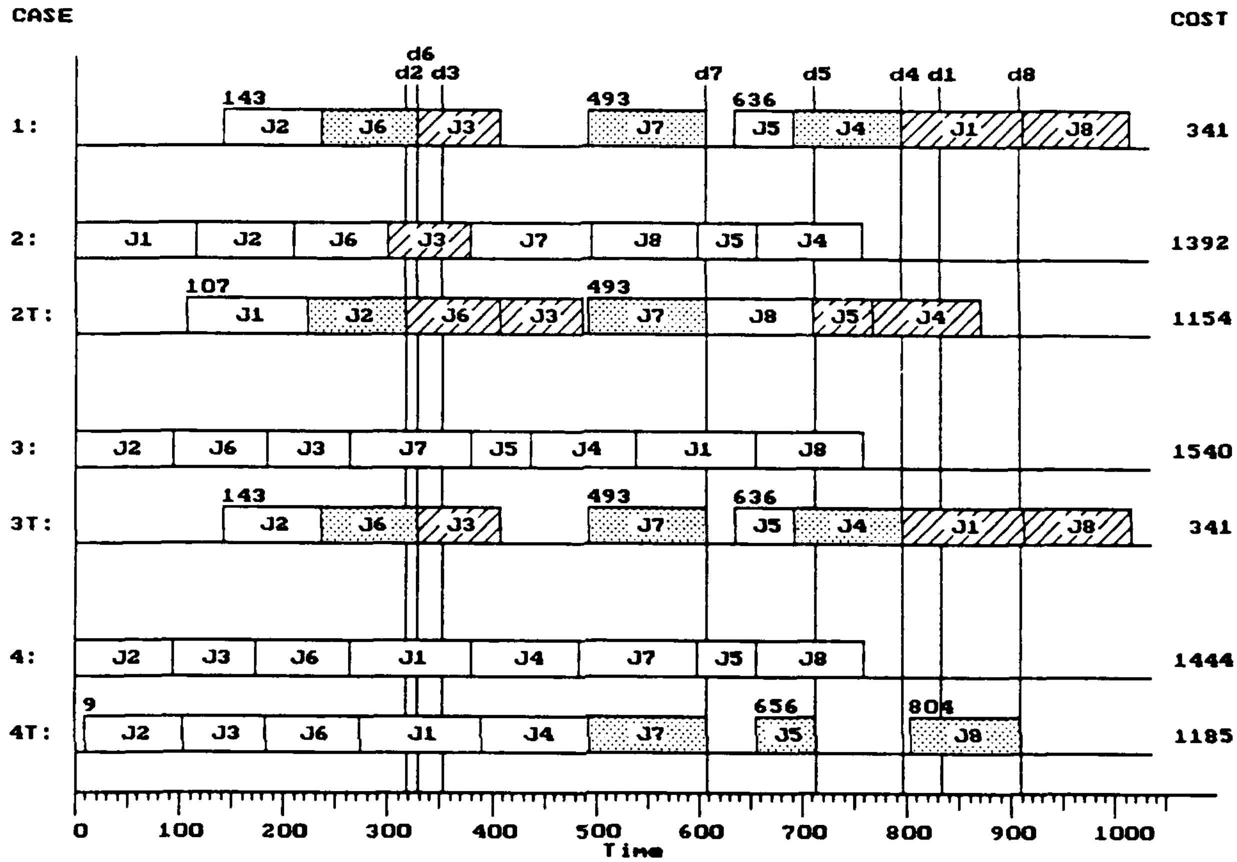


Figure 8. Gantt Charts for the Solutions to Baker's Modified Problem 15

Case 1: Optimum solution

Case 2: Optimum solution when idle time is not allowed
Case 2T: Case 2 solution after applying TIMETABLER

Case 3: Earliest Due Date (EDD) with no idle time
Case 3T: Case 3 solution after applying TIMETABLER

Case 4: Minimum tardiness solution
Case 4T: Case 4 solution after applying TIMETABLER

identical to the optimum schedule (Case 1). Case 4T also had three idle periods and was the only timetabled schedule to have no tardy jobs.

Results with Reduced Earliness Penalties

Table V summarizes our results for the same Baker problems with the additional modification: e(i)=.25, t(i)=1, for each job i (see Expression 1). This made the objective function for these problems the same as that studied by Ow and Morton (1989) In viewing these results, the same general observations can be made as when e=t=1. However, since the earliness penalties are much lower, the magnitude of the differences between schedules is reduced.

Comparison with Ow and Morton's Results

An important distinction between our work and that reported by Ow and Morton (1989) is that in their study of the ET problem, they assumed that no idle time is allowed in the schedule. As our results show, this requirement can lead to substantially sub-optimum schedules. In their work, Ow and Morton studied a number of heuristic scheduling methods. Although we did not solve the problems in this paper using any of their algorithms, we can make some statements about the relative performance of any such heuristic algorithm used in solving these problems. Any such algorithm will perform no better than the Case 2 solution noted here. This follows since Case 2 is the optimum solution to the problem when idle time is not allowed. However, subsequent application of TIMETABLER can alter this conclusion. The notable illustration of this is the performance of EDD after it is timetabled with TIMETABLER. Before timetabling, Case 3, as expected, is never better than Case 2. After timetabling, the situation becomes drastically different. An interesting question is whether other heuristic methods such as Ow and Morton's beam search would behave more like EDD or more like the Case 2 solution. Ow and Morton's results for their problems indicate their solutions to be closer to the Case 2 solutions than to the EDD solutions. (See Table 1 of Ow and Morton [1989]). They found EDD to be disappointing in comparison to the other heuristics they tested. Our results are similar. But our results show EDD after timetabling to be surprisingly good and it may well be better than any of Ow and Morton's heuristics. A more thorough evaluation of the "EDD after timetabling" heuristic appears to be a worthy topic for future research. Likewise, an interesting issue for future research might be to investigate whether heuristic search approaches such as those proposed by Ow and Morton can be enhanced by integrating TIMETABLER into the search procedure.

Table V
Additional Computational Results
for Baker's Problems Modified Such That Tau=0.2
When the Cost Function Coefficients are e=0.25, t=1, for Each Job

		COST	% ABOVE OPTIMUM					
Baker Problem	Range Factor	Case 1	Case 2	Case 2T	Case 3	Case 3T	Case 4	Case 4T
4	0.087	413.5	0.00	0.00	10.52	10.52	27.39	27.39
12	0.101	438.5	0.00	0.00	19.50	19.50	7.75	7.75
10	0.111	644	0.00	0.00	10.40	10.40	3.69	3.69
8	0.116	550.25	0.00	0.00	9.36	9.36	8.63	8.63
14	0.166	516.25	2.03	0.00	8.43	8.43	2.52	2.52
16	0.167	594.5	0.00	0.00	3.03	3.03	4.12	4.12
2	0.197	378.5	4.36	0.79	26.09	22.52	22.85	22.85
6	0.207	391	0.00	0.00	18.09	15.03	22.70	22.70
5	0.412	329	23.48	0.00	35.64	5.32	36.93	25.53
7	0.452	315	5.95	0.00	15.95	10.00	15.79	15.79
1	0.543	233.25	38.16	0.00	59.70	21.44	47.37	13.29
9	0.638	212	62.97	30.54	88.33	20.28	84.55	22.88
13	0.653	196	98.47	56.38	137.88	16.33	117.60	37.50
3	0.667	140.75	122.02	96.80	154.00	2.49	131.44	106.22
11	0.696	100.75	180.15	121.09	238.71	0.00	235.73	227.79
15	0.778	122.5	189.59	118.57	214.29	9.39	194.69	141.84

Case 1: Optimum solution

Case 2: Optimum solution when idle time is not allowed
Case 2T: Case 2 solution after applying TIMETABLER

Case 3: Earliest Due Date (EDD) with no idle time
Case 3T: Case 3 solution after applying TIMETABLER

Case 4: Minimum tardiness solution
Case 4T: Case 4 solution after applying TIMETABLER

The relatively poor performance of the schedules of Case 4 and 4T (see Tables IV and V) confirms Ow and Morton's conclusion that it may be expensive to ignore early costs in the search for a solution to the ET problem. Ow and Morton also concluded that the ratio of e/t had little influence on the nature of their results. Our experience with the two problem sets (e=1, t=1; and e=.25, t=1) led to the same conclusion.

CONCLUSION

Baker (1974, p.2) has defined scheduling as "... the allocation of resources over time to perform a collection of tasks." The phrase "allocation over time" implies that scheduling can be subdivided into the two activities -- sequencing, and timetabling. Sequencing is specifying the order (permutation) in which the tasks are to be performed. Timetabling is the assignment of start and completion times to each task in the permutation. This report has highlighted the importance of timetabling as an integral part of scheduling.

Referring again to Figure 2, we can give a geometric interpretation to our results. For the general case of n jobs we can think of an n-dimensional solution space with each dimension corresponding to the completion time of a particular job. The feasible set of solutions will be comprised of n! disjoint regions, each one corresponding to a particular permutation of the n jobs. Each of these regions defines a convex set thus permitting the definition of a convex objective function over the region. Within each region, any solution can be defined in terms of a base solution and a set of local shifts (vectors). We have provided a simple (polynomial time) procedure for finding an optimum solution to this function given a priori specification of the region.

The class of problems which we define as n/1//convex encompasses a larger class of performance measures than the regular measures. Yet, we have shown that even for non-regular measures, it is not much more difficult to design an enumerative search than it would be for an instance of n/1//convex with a regular measure. Our preliminary computational results show that problems of up to 8 jobs can be solved in reasonable time on a computer with little attention given to the development of special bounding methods and/or dominance checks. The intriguing finding here is that applying TIMETABLER within the search process can lead to significantly different schedules and costs. A desirable avenue of further research would be to perform an in-depth computational study of using TIMETABLER with different non-regular objectives such as lateness variance or

weighted ET, and to test the sensitivity of these results to different problem input parameters. We hope that our findings will encourage such further research into this important class of problems.

In connection with such a study it is the authors' opinion there is a clear need to re-examine the design of the parameters used in characterizing the problems employed in computational studies. For example, for problems when the objective is convex but non-regular, the Tau parameter may not be as meaningful as it is when regular performance measures prevail. Likewise, there is a need to characterize problem instances in terms of more universally-recognizable parameters such as machine utilization and to extend the problem specification to include other factors such as ready times and set-up times and to review the choice of underlying distributions for input such as due dates and processing times. Moreover, it would be a service to the research community to precisely specify exactly how such problem sets are generated so that future research efforts could independently confirm earlier results and allow for more readily-comparable statements between studies. Freed of the tedium of sample problem construction, future researchers could concentrate on more important issues such as algorithm design and comparison.

FOOTNOTE

(*) Personal conversation with Baker clarified that the 16 subject problems came out of the Baker and Martin work and that in fact the sample problems were constructed by randomly generating due dates from two distributions: one with mean Tau = 0.2, the other with mean Tau = 0.6, so that the procedure of Ow and Morton (1989) closely followed that of Baker and Martin (1974).

Appendix A -- The Preemption Issue

In the context of scheduling, the term preemption refers to the practice of deliberately interrupting the processing of a job. Processing of the interrupted job commences again after the completion of any interrupting activities. In many scheduling applications it is important to consider solutions which employ preemption. But in such cases there are often practical limitations on how preemption is to be deployed. More specifically, there may be practical bounds limiting the minimum time that a job is to occupy a machine before it is permitted to be interrupted. For example, when a job is comprised of a batch of identical discrete parts, it may make sense to limit preemption so that interruptions would not be scheduled to occur in the midst of processing an individual

part. Similarly, we can imagine cases where the transfer container size may render it practical to bound the minimum number of items to be completed before interruption is permitted. Moreover, there may be cases when shop practice renders an effective practical lower limit on a production batch size. For example, it might be judged that machine setups may occur with a minimum frequency as in a factory with interruptions permitted no more frequently than once per production shift. Maxwell and Muckstadt (1985) refer to this minimum period as a _base planning period_ so that all reorder intervals (and therefore all production batch sizes) must be an integer multiple of this period.

Assume that there are practical limitations on the minimum allowed production batch sizes. Then in developing solution procedures for such scheduling problems we can assume without loss of generality that preemption is not allowed. This is so because any instance of a problem which allows preemption can be converted in polynomial time to an equivalent problem which does not allow preemption in the following manner. First partition each job into job steps (atomic units), which for practical reasons may not be preempted. Then relabel the job steps as jobs, and revise the setup time matrix as needed. This argument must surely be well-known. It is nevertheless important to note because it extends the argument presented by Conway, Maxwell, and Miller (1967, p. 24), that it is unnecessary to consider schedules with preemption when a regular performance measure is involved. The "atomic" reduction described above shows that the regular performance measure assumption is not necessary; i.e., preemption need not be explicitly considered _regardless_ of the performance measure.

Appendix B -- TIMETABLER Provides a Lower Bound

Theorem 4. The cost of a partial schedule produced by TIMETABLER is a lower bound on the cost of a schedule generated from it.

Proof. Let S(t,P) be the schedule produced by TIMETABLER, and P' be the set of unscheduled jobs. Assume the contrary, that the cost of S(t,P) is not a lower bound. Thus, there exists some schedule S(q,TP), q > t, having a lower cost than S(t,P). Since the scheduling of the jobs in T could not contribute a negative cost, the (re)timetabling of the jobs in P must have resulted in a lower cost than in the S(t,P) schedule. But this retimetabling could have involved only shifting of jobs (i.e. no change in the permutation P). If such shifting reduced cost, it is a contradiction of Theorem 2, which says that S(t,P), the product of TIMETABLER, is a semi-active schedule. Q.E.D.

III. THE CLEMSON QUICK RESPONSE PLANNER

In a manufacturing environment, a machine can perform various different types of operations. At any point in time, it can have a number of outstanding jobs waiting in queue, each job requiring one of the different operations that can be performed by the machine. Each job could have a different priority and so a different penalty for not completing at its due date. In such a situation, the manager would like to see all the job information, machine utilization, etc., and try to schedule jobs so that the least penalty is incurred. Although all the information may be somewhere in a database it may be difficult to interpret it in a reasonable amount of time. Hence a graphical display of information would be useful for scheduling and decision making. The Quick Response Planner system addresses the aforementioned needs.

The Quick Response Planner (QRP) is a prototype single machine scheduling system with a graphical interface for inserting, scheduling, and manipulating orders and looking at schedules via Gantt charts. It is in essence an electronic leitstand. As defined by Adelsberger and Kanet (1991), a leitstand is a computer aided decision support system for interactive production planning and control. A leitstand is comprised of a graphics component, a schedule editor, a schedule evaluator, an automatic schedule generator, and a database manager.

The QRP software has been written in C and uses X Windows and Motif libraries. It also uses the Motif's User Interface Language for widget (graphical object) definitions. The advantage is that the user can change the characteristics of the widgets without recompiling the entire program. Therefore a manager who adopts QRP can easily customize it to handle the needs of a particular organization.

The initial work was started in OS/2 and used Presentation Manager. Later, it was ported and enhanced under the AIX/X/Motif environment.

HARDWARE

The present hardware requirements for QRP is an IBM PS/2 80386 machine with a 120 MB hard drive and 8 MB of main memory and an 8514 monitor with an 8514A adapter card. It has also been ported to the Sun 4.0 workstation. A color monitor is essential.

SOFTWARE

QRP runs under AIX version 1.1 and X Windows version 1.0 and OSF/MOTIF version 1.0. However, it can run under any system which has X Windows and Motif installed properly. It can also run under

Sun OS/2 with Motif 1.1 and has been successfully ported to that environment with some limitations.

FUNCTIONS OF QRP

This section describes the different functions performed by QRP.

Automatic Schedule Generator

One of the foremost features of the system is to generate a schedule automatically using one of the several embedded algorithms. For generating schedules two steps have to be followed: sequencing and timetabling. First the jobs have to be sequenced. Sequencing can be accomplished manually or by using one of the built-in algorithms: SPT, LPT, FIFO, EDD or MDD. Timetabling can be accomplished manually or one of these algorithms can be used: ASAP, ALAP or TIMETABLER. A description of these algorithms is provided later.

Schedule Evaluator

QRP uses five performance measures to evaluate a schedule. They include cost, tardiness, flowtime, resource utilization, and average inventory. To understand each of these, first we need to define the various characteristics of a job. Although a job can have various information associated with it, the ones found useful for our purposes are as follows.

Order No.	The order no. associated with this job.
Customer No.	The customer no. of the customer who placed this order.
Due Date	The due date of the order as specified by the customer.
Ready Date	The time when the raw materials or the predecessor operations will be ready. The job can start only after this date.
Completion Date	The time when the job is completed. This is only available after the job has been scheduled.
Early Penalty	The penalty (per unit time period) incurred if the job is finished earlier than its due date.
Late Penalty	The penalty (per unit time period) incurred if the job is finished later than the due date.
Quantity	The number of units needed to be produced.

Product Type	The type of product that is to be produced. Even though they may have some differences, products of the same type can be processed in sequence without incurring setup time. From the type, one can determine the per unit processing time of the job.
Processing Time	This is the total processing time for the job. It is the product of unit processing time (which is determined by the product type) and the quantity to be produced.
Setup Time	The time required to reset the machine to handle a job of a different type than the previous job.

Schedule performance measures are defined as follows.

a) Cost - The cost of a schedule is defined as the sum of the early and late penalties of each job in the schedule. There is no penalty if the job finishes at its due date. More precisely, the penalty for job j is defined as follows:

```
if C[j] > D[j]
                cost[j] = (C[j]-D[j])*LP[j]*Qty[j]
else
                cost[j] = (D[j]-C[j])*EP[j]*Qty[j]
```

where EP[j], LP[j] represent the early and late penalties, and Qty[j] represents the quantity of items in job j, and C[j] and D[j] are the completion date and due date for job j, respectively.

b) Tardiness - The tardiness of a schedule is the total tardiness of all the jobs in the schedule. The tardiness of each job j is defined as follows:

```
if C[j] > D[j]
                tardiness[j] = (C[j]-D[j])*Qty[j]
else
                tardiness[j] = 0
```

c) Flowtime - Flowtime is the total time a job spends on the shop floor. It is the difference in time when it starts processing and when it leaves the workcenter. The flowtime of a job is calculated as follows:

```
if C[j] > D[j]
```

```
                    flowtime[j] = (C[j]-S[j])*Qty[j]
     else
                    flowtime[j] = (D[j]-S[j])*Qty[j]
```

where S[j] represents the start time for job j in the schedule.

There are two assumptions being made here:

1) The raw material for the job is accepted at the start time and not the ready date.
2) The order is shipped at the due date if the completion date is earlier than the due date, else it is shipped at its completion date.

d) Resource Utilization - The resource utilization is defined as the ratio of the amount of time the machine is busy processing jobs (setup time included) to the total time the machine is available. The total time the machine is available is defined as the difference between the completion date of the last job in the schedule and the start of the first job.

e) Average Inventory - This is defined as the average number of units (of finished product) present in inventory. An assumption is made here that if a job completes after its due date then the job is shipped immediately. So if there are two jobs, J1 and J2, in the schedule as follows

	Start Date	Completion Date	Due Date	Quantity
J1	4	10	12	2
J2	7	15	11	3

then the average inventory would be calculated as follows:

Avg. Inventory = ((12-4)*2+(15-7)*3)/(15-4)

Schedule Editor

Once a schedule has been fixed the user might want to move some jobs around so that they start at different times. For this kind of editing on the screen, the mouse can be used to change the start times of jobs. The editing capabilities provided are moving, pushing, splitting, and joining of jobs. "Move" option lets the user move one particular job at any point in time where the job fits. "Push" option pushes the job left or right and when the job bumps against another job, that job is pushed too. The "Split" option lets the user split the job into two parts. The "Join" option joins two consecutive jobs of the same order which had been split earlier.

Graphics Component

The graphics component consists of showing the jobs in the traditional Gantt chart, called the 1D Gantt chart, and in 2D, which shows the desirable region for scheduling jobs. The desirable region is defined as the region between the ready date and the due date. A timescaling and scrolling bar is provided to allow looking at varying lengths of time. The job information can be seen at the bottom of the chart. The performance evaluators are shown at the top of the chart. The system is completely menu driven and uses dialogue boxes, selection boxes, etc., for input from the user.

DESCRIPTION OF A SAMPLE SCREEN

The introductory screen for QRP is shown in Fig. 9. A sample of the working screen is shown in Fig. 10. There are basically five parts to the working screen.

Part a)

This part shows the performance measures, namely, Cost, Tardiness, Flowtime, and Resource Utilization. The Present, Previous, and Best stand for schedules, i.e., the Present Schedule is the one currently displayed. The Previous Schedule is the one most recently saved to a file, and the Best Schedule is the one with the best score on a certain performance measure (which has been saved to a file). Values of these measures are given in numerical form and are also represented graphically as bar charts.

Part b)

This is the time scale and the scrolling bar. The length of the black rectangular bar is the time horizon shown on the Gantt chart in Part c. The time horizon is the total length of time that can be seen on the Gantt chart. To change the length of the bar, the left mouse button has to be pressed near the right end of the bar; keeping the mouse button pressed, the size can be changed. "Rubberband" techniques are used to change the rectangle size while the mouse button is being dragged. To move the rectangle, the mouse button has to be clicked at the desired start time.

Part c)

This is a traditional Gantt chart. The rectangles represent jobs. The length of the black solid filled rectangles represent setup times. The length of a shaded bar represents the length of time the job is to run on the machine. The beginning and end of the shaded rectangles represent the start and end time of the jobs. Jobs of the same shade are jobs of the same operation. Jobs of the same

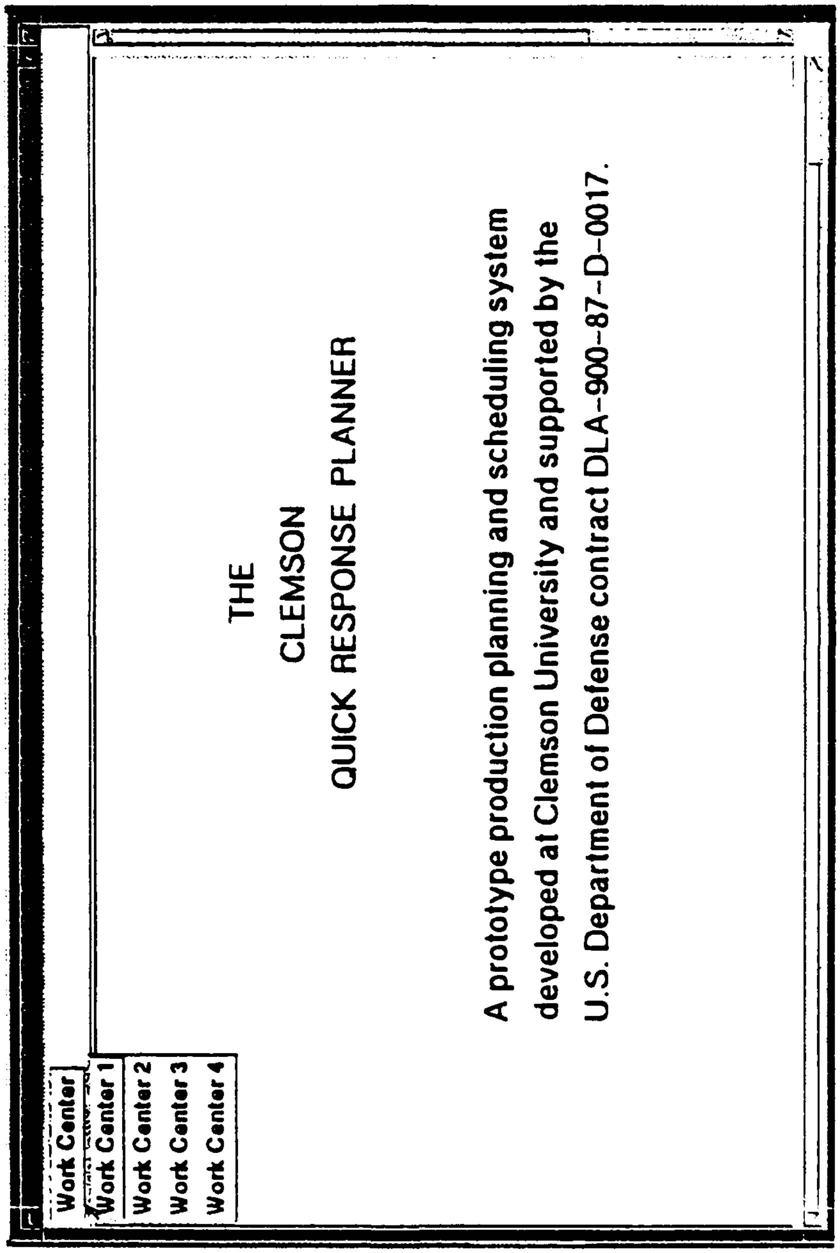

Work Center
Work Center 1
Work Center 2
Work Center 3
Work Center 4
THE
CLEMSON
QUICK RESPONSE PLANNER
A prototype production planning and scheduling system
developed at Clemson University and supported by the
U.S. Department of Defense contract DLA-900-87-D-0017.

Figure 9

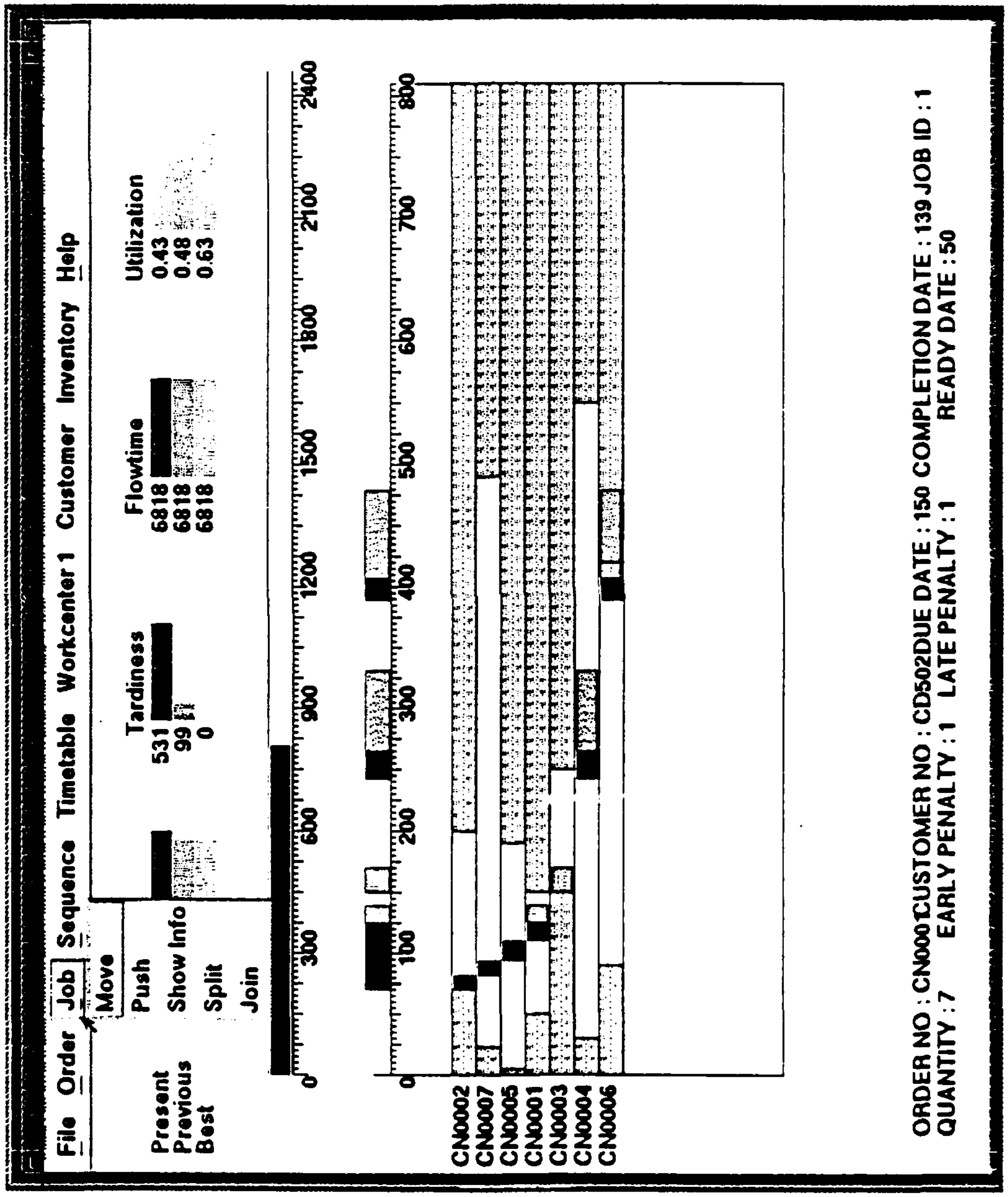

File Order Job Sequence Timetable Workcenter 1 Customer Inventory Help
Move
Push
Show Info
Split
Join
Present
Previous
Best
Tardiness
531
99
0
Flowtime
6818
6818
6818
Utilization
0.43
0.48
0.63
0 300 600 900 1200 1500 1800 2100 2400
0 100 200 300 400 500 600 700 800
CN0002
CN0007
CN0005
CN0001
CN0003
CN0004
CN0006
ORDER NO : CN0001CUSTOMER NO : CD502DUE DATE : 150 COMPLETION DATE : 139 JOB ID : 1
QUANTITY : 7 EARLY PENALTY : 1 LATE PENALTY : 1 READY DATE : 50

Figure 10

operation have the same machine setup time. That is, if there are two consecutive jobs of the same operation, the latter will have no setup time. The jobs are actually color coded on the screen, but are shown in different shades in the black and white figures.

Part d)

Like Part c) this is also a Gantt chart but it displays additional information. The text shown on the left is the order number. The ready date and the due date for each job are shown graphically. The end of the shaded rectangle on the left is the ready date. The start of the shaded rectangle on the right is the due date for that job. The unshaded region is the time wherein one would normally like to schedule the job. The colored rectangles with an initial black rectangle are the jobs, where the length of the colored rectangle denotes the actual processing time of the job, while the black rectangle represents the setup time.

Part e)

This part shows selected information for each job. This feature can be invoked by selecting the Show Info option in the Job menu and then selecting any job with the mouse.

MENU OPTIONS

File

The File menu options perform various functions such as deleting a file, saving a file, saving as another file, opening a file, etc. Only those files should be selected that have an extension of "dbl" as they are the binary files containing schedule information. Trying to open a file with a different extension will result in an error. The submenus are discussed below. The file menu is shown in Fig. 11.

<u>New</u>: Clears the present screen and initializes the program. If changes had been made after the last save was done on the current set of jobs then the user is prompted to save the current schedule.

<u>Open</u>: Opens a file, which has a schedule stored in it. A file selection box comes up giving the list of all the files (Implementation note: files containing schedules have an extension of "dbl"). Selecting any of them puts that schedule on the screen.

<u>Save</u>: Saves the current schedule in the same file that had been opened using the Open option.

<u>Save As</u>: Same as Save, except that it will save the schedule in a different file than what had been opened. A dialogue box appears to prompt the user to enter the file name.

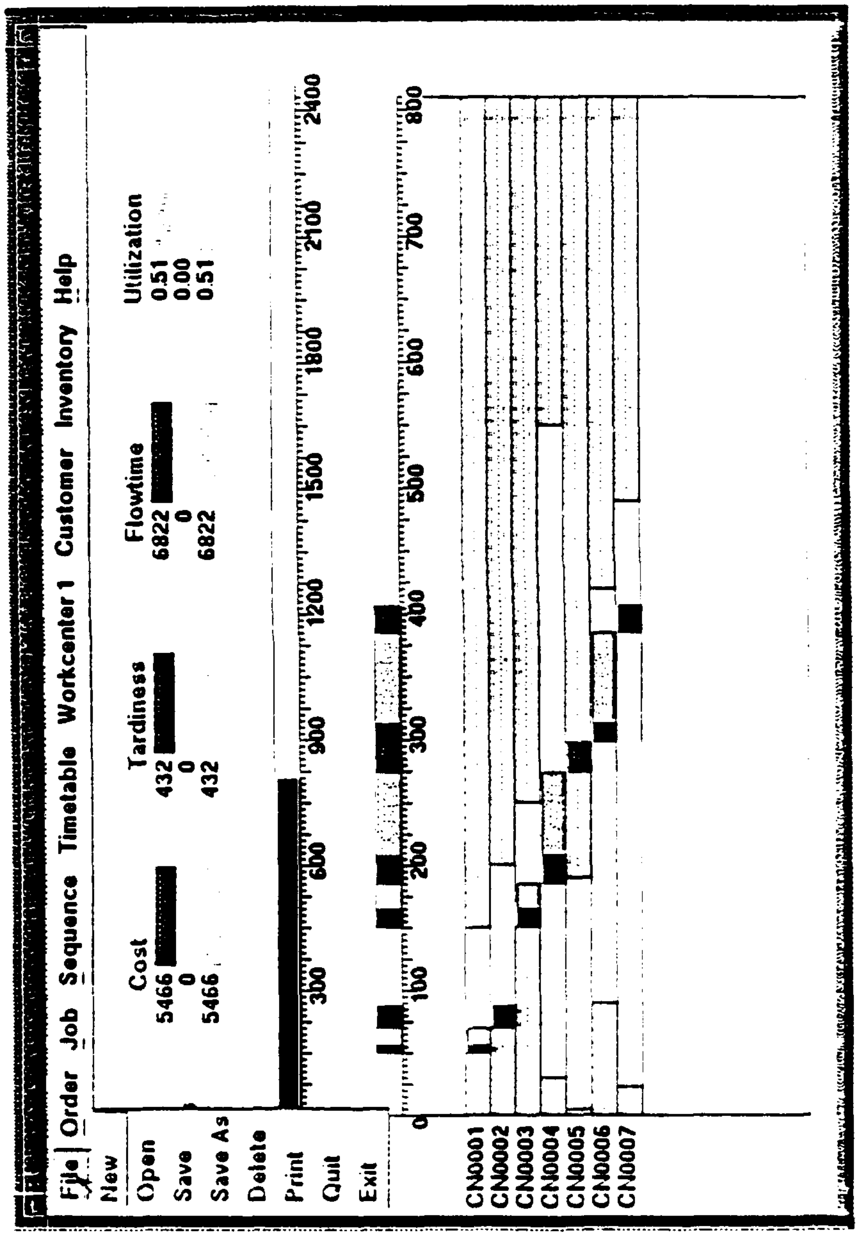

File
Order
Job
Sequence
Timetable
Workcenter 1
Customer
Inventory
Help
New
Open
Save
Save As
Delete
Print
Quit
Exit
Cost
5466
0
5466
Tardiness
432
0
432
Flowtime
6822
0
6822
Utilization
0.51
0.00
0.51
CN0001
CN0002
CN0003
CN0004
CN0005
CN0006
CN0007

Figure 11

Print: Prints the contents of the window. The shape of the cursor changes and then the user can click on the window he wants to print. If QRP is running on an IBM PS/2, the printer should be an HP LaserJet with 2MB memory, connected on the parallel port.

Delete: Deletes the current file (the file from which the current schedule was retrieved).

Quit: Displays the introductory screen (shown in Fig. 9).

Exit: Exits from the system after confirming. A dialogue box will appear for the user to confirm that an exit is desired.

Order

An Order has basically the same characteristics as the job except that an order can be split into many jobs each of which handles part of the order. This menu item provides the interface for performing various operations on the orders. A template appears on the screen to enter an order. A sample screen is shown in Fig. 13. The Order menu is shown in Fig. 12.

Insert: The template shown in Fig. 13 appears on the screen. Entering the different parameters and then clicking on "schedule" will result in the scheduling of the order.

Delete: The shown template in Fig. 13 appears. Using the "next" and the "prev" options, the different jobs in the current opened file can be selected. Clicking on the Delete button will delete the selected job.

Modify: The template shown in Fig. 13 appears. Any of the order parameters can be changed and the associated job information will be modified.

Show Info: The template appears. The parameters cannot be changed. They can only be viewed. This feature is implemented by setting the XmNedit resource of the text widget to False. This prevents the user from accidentally changing any information.

Job

Various operations can be performed on a job using this software. A job is a single entity capable of running independently on a machine. Jobs are represented as rectangles on a Gantt chart. (Implementation note: To keep track of which job is selected, the mouse movements are continuously tracked and based on the visible jobs a job is selected if the mouse coordinates matches with the job coordinates. The rectangles are drawn using Xlib calls. Instead, they could have been represented by Pushbutton widgets. But since widgets consume a lot of memory it was decided to use basis Xlib calls.) The Job menu is shown in Fig. 14.

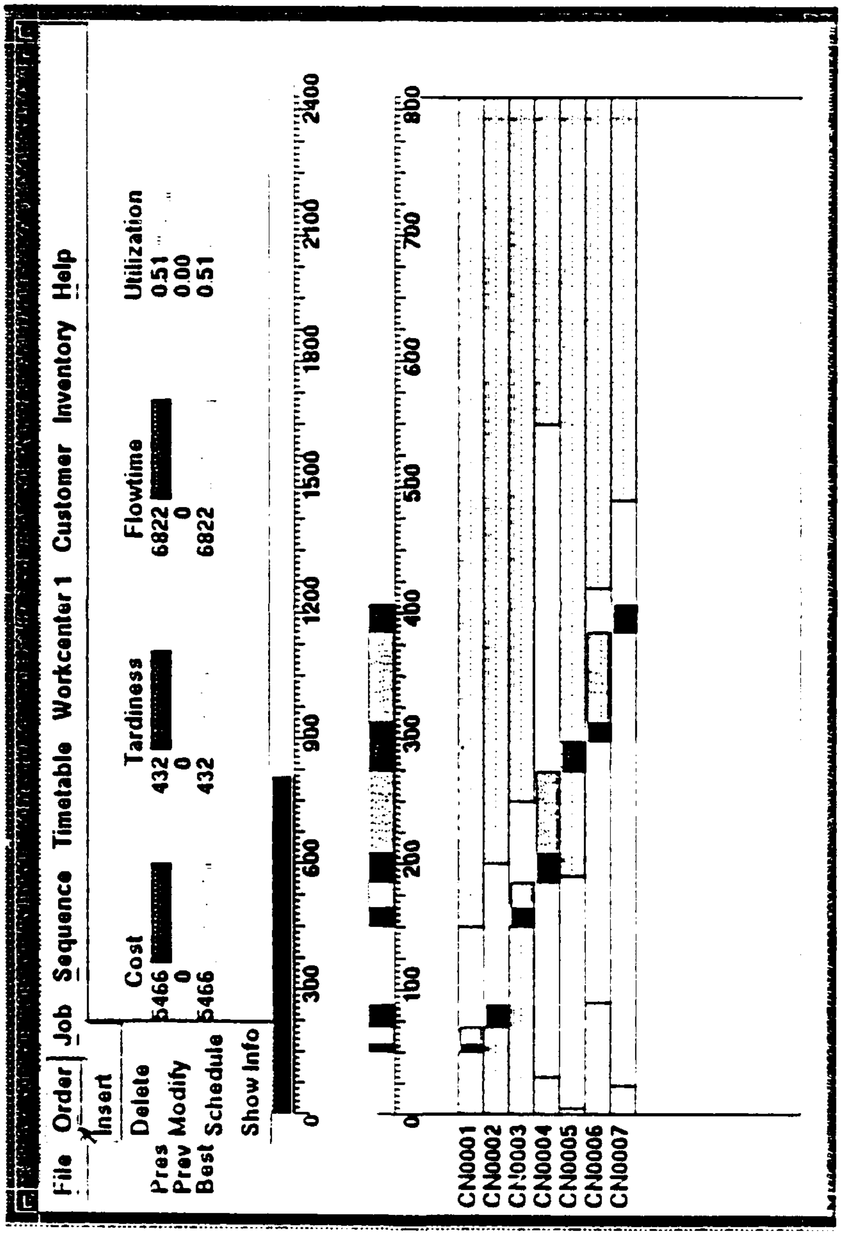

File Order Job Sequence Timetable Workcenter 1 Customer Inventory Help
Insert
Delete
Modify
Schedule
Show Info
Cost
Tardiness
Flowtime
Utilization
Pres 5466 432 6822 0.51
Prev 0 0 0 0.00
Best 5466 432 6822 0.51
CN0001
CN0002
CN0003
CN0004
CN0005
CN0006
CN0007

Figure 12

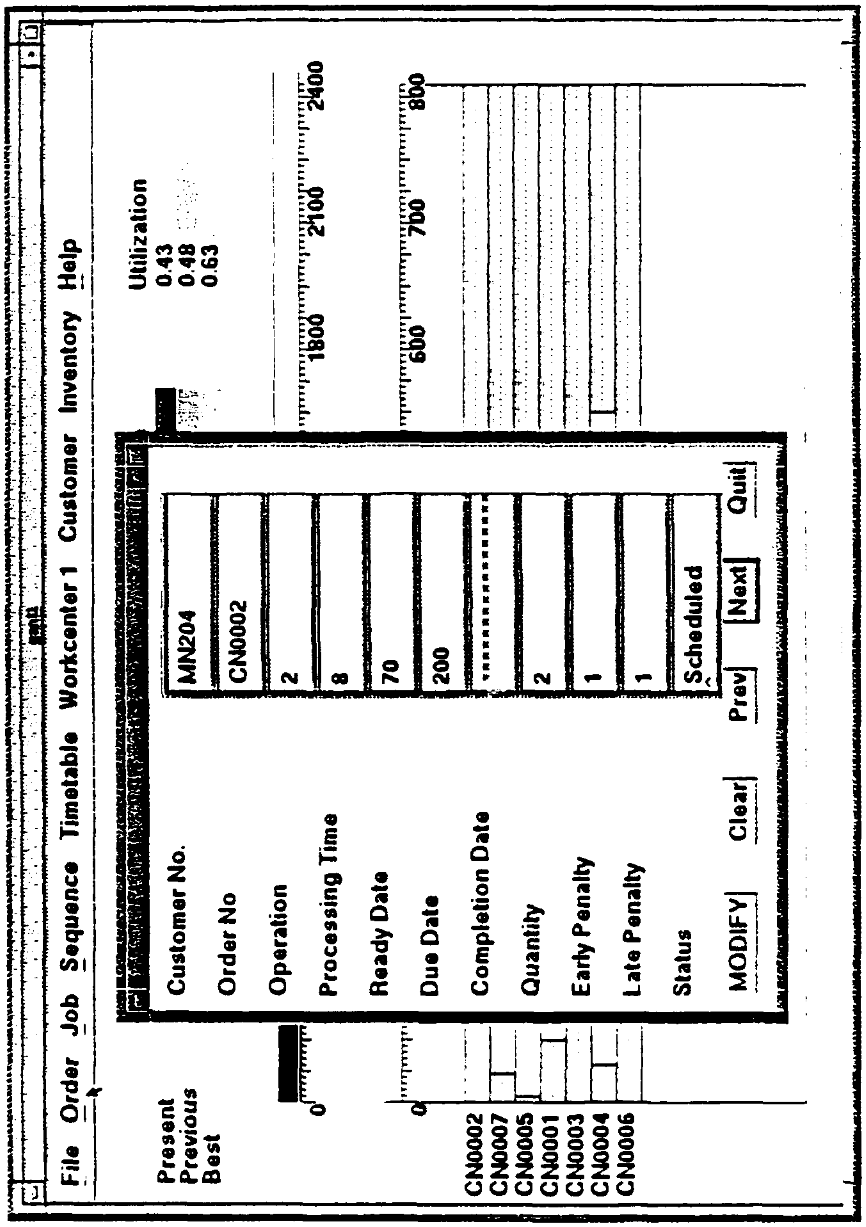

File
Order
Job
Sequence
Timetable
Workcenter 1
Customer
Inventory
Help
Present
Previous
Best
Utilization
0.43
0.48
0.63
0
1800
2100
2400
0
600
700
800
CN0002
CN0007
CN0005
CN0001
CN0003
CN0004
CN0006
Customer No.
MN204
Order No
CN0002
Operation
2
Processing Time
8
Ready Date
70
Due Date
200
Completion Date
Quantity
2
Early Penalty
1
Late Penalty
1
Status
Scheduled
MODIFY
Clear
Prev
Next
Quit

Figure 13

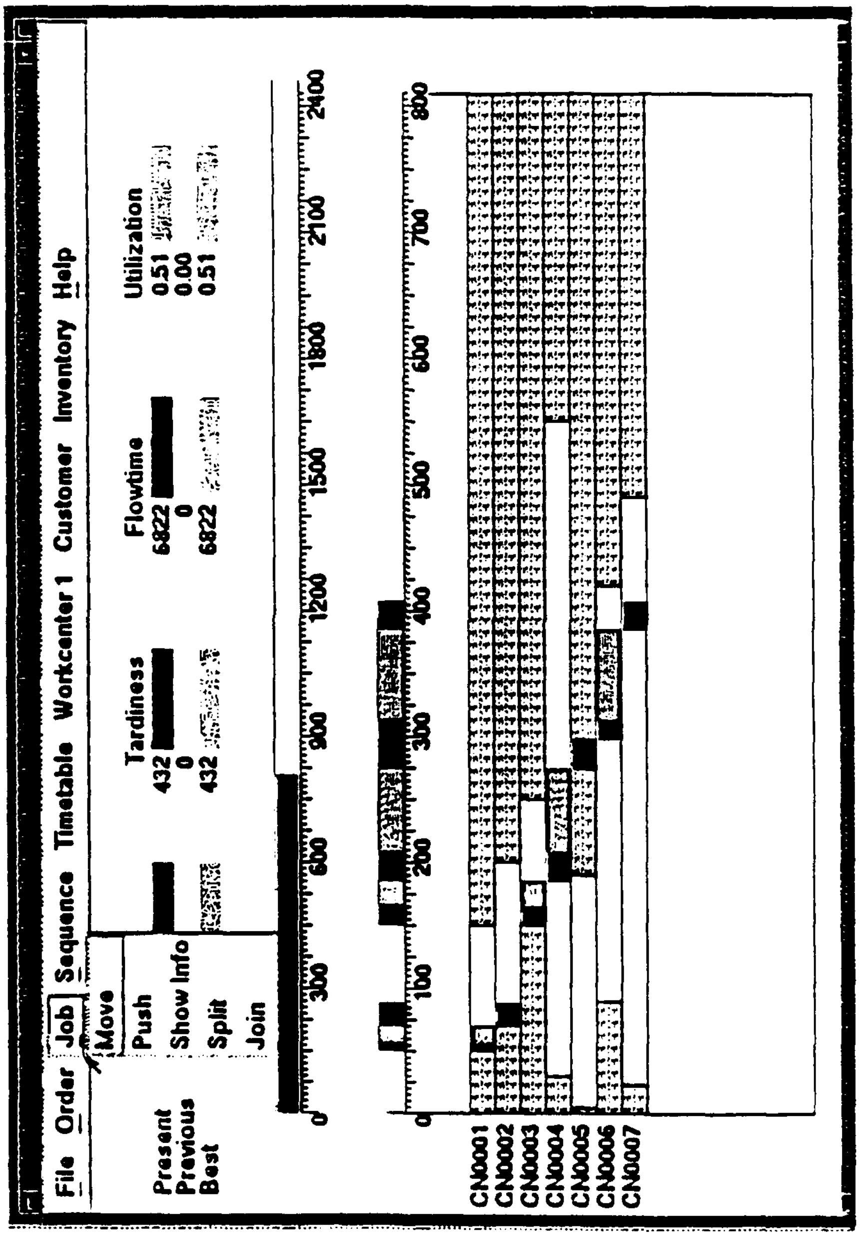

File
Order
Job
Sequence
Timetable
Workcenter 1
Customer
Inventory
Help
Move
Push
Show Info
Split
Join
Present
Previous
Best
Tardiness
432
0
432
Flowtime
6822
0
6822
Utilization
0.51
0.00
0.51
CN0001
CN0002
CN0003
CN0004
CN0005
CN0006
CN0007

Figure 14

Move: Reschedules a job. The job can be selected by selecting the job rectangle with the mouse The job then moves along with the mouse. Clicking the mouse again (where the job is to be placed) draws the job again in the new position. (Implementation note: The rubber-band technique is used. The right parameters are set in GC, GCxor as the copy type, so that a line drawn again between two coordinates turns the pixels on and off.)

Push: Pushes a job or a set of jobs either left or right. When a job bumps against another job that job is pushed too. (Implementation note: For drawing the rectangles the XCopyArea call is being used.)

Split: Splits a job into halves. After this option has been selected, moving the cursor near the jobs will make a vertical line appear across the jobs. Clicking on a job would result in the job being split into two parts.

Show Info: Show job information. Clicking on the job would result in the job information being shown at the bottom of the window.

Sequence

Various sequencing rules can be used to schedule the jobs. The ones available are outlined below. The menu is shown in Fig. 15.

SPT: ("Shortest Processing Time") Sequences jobs in order of increasing processing time.

LPT: Sequences jobs according to longest processing time first.

FIFO: Sequences jobs according to first in first out.

EDD: Sequences jobs according to earliest due date first.

MDD: Sequences jobs according to modified due date first. (It is an algorithm developed by Baker and Kanet, 1983.)

OPT: Sequences jobs in an optimal way by applying a branch and bound algorithm.

Best: Retrieves the best schedule (which has been saved in a file) for the present set of jobs according to the various performance measures shown on the top of the chart. The subchoice "tardiness" retrieves the schedule that had the least tardiness for the current set of jobs. The subchoice "cost" retrieves the schedule that had the least cost for the current set of jobs. The subchoice "flowtime" retrieves the schedule that had the least flowtime for the current set of jobs. The subchoice "utilization" retrieves the schedule that had the maximum utilization for the current set of jobs.

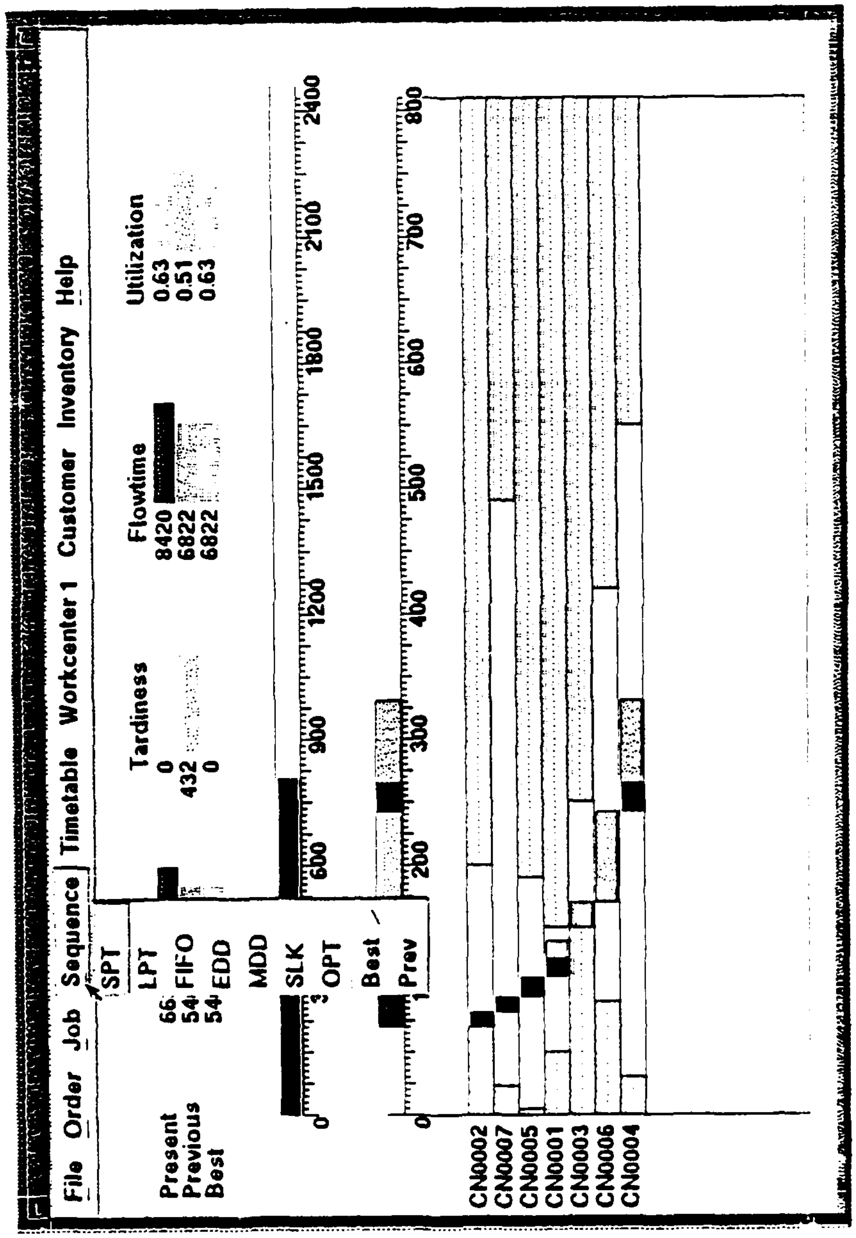

File Order Job Sequence Timetable Workcenter 1 Customer Inventory Help
SPT
LPT
FIFO
EDD
MDD
SLK
OPT
Best
Prev
Present
Previous
Best
Tardiness
0
432
0
Flowtime
8420
6822
6822
Utilization
0.63
0.51
0.63
CN0002
CN0007
CN0005
CN0001
CN0003
CN0006
CN0004

Figure 15

Previous: Retrieves the most recently saved schedule.

Timetable

This option generates schedules once the sequence has been determined. Several timetabling options (algorithms) are available as outlined below. The menu is shown in Fig. 16.

ASAP: Schedules jobs in order of increasing ready date ("as soon as possible").

ALAP: Schedules jobs as late as possible.

TIMETABLER: Schedules jobs according to the TIMETABLER algorithm developed by Davis and Kanet (1991) and described in section II of this report. This option results in the lowest cost schedule possible, but may require excessive processing time for schedules of more than 7 jobs.

Workcenter

This menu provides the interface for inserting, changing, modifying, and showing workcenter information. The information for each workcenter consists of the following items. The menu is shown in Fig. 17.

1. Workcenter name
2. Total number of operations that can be performed.
3. Type of different operations and the unit processing and a setup time matrix which shows the time for each.

Customer

This menu provides the interface for inserting, modifying, showing, and deleting customer information. The different fields are self-explanatory. A sample template is shown in Fig. 18. The menu is shown in Fig. 19.

Insert: A template appears as shown in Fig. 18. The various customer record fields can then be entered. (Implementation note: as explained earlier, the same hierarchy of widgets is maintained in showing the template.)

Modify: The same template shown in Fig. 18 appears. Any of the fields can be modified.

Delete: The same template shown in Fig. 18 appears. Clicking on the "Delete" button will delete that particular record.

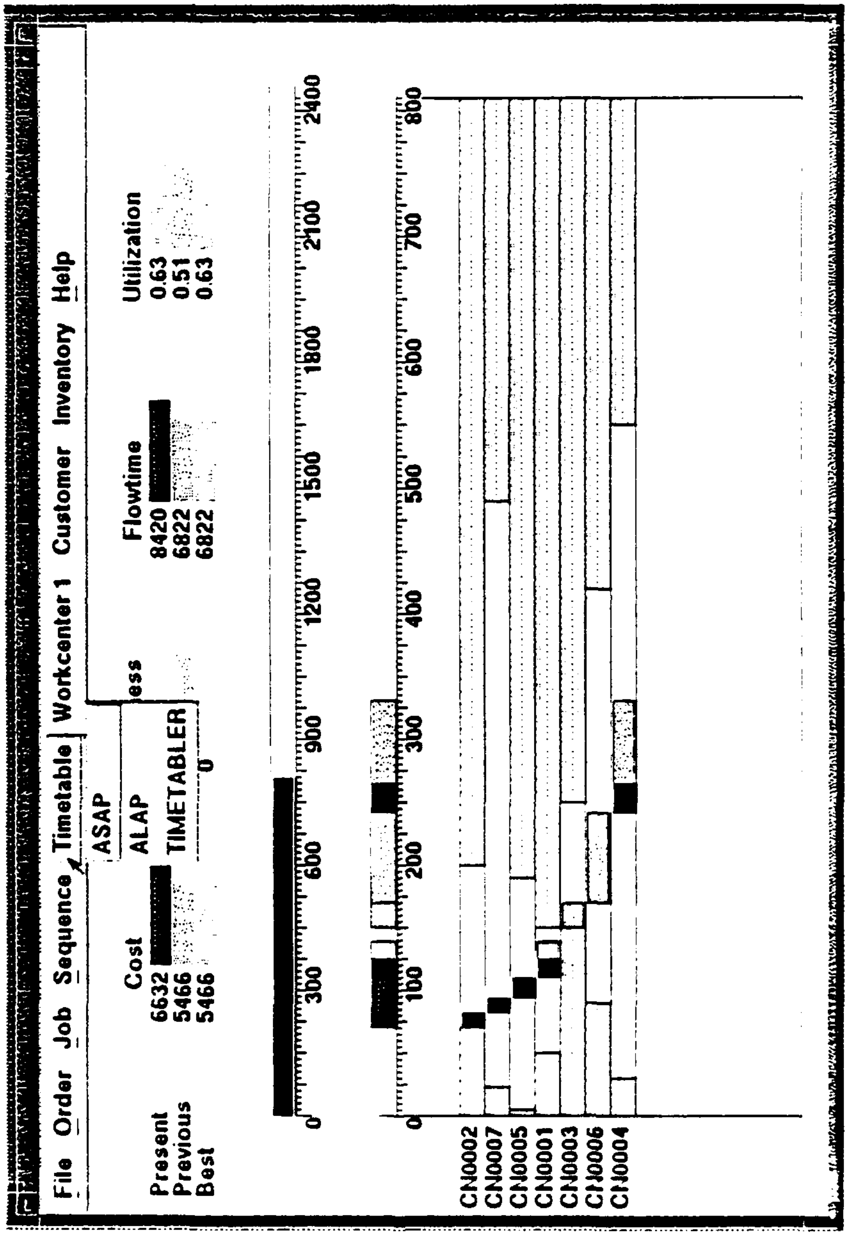

File Order Job Sequence Timetable Workcenter 1 Customer Inventory Help
ASAP
ALAP
TIMETABLER
Cost
Flowtime
Utilization
Present 6632 8420 0.63
Previous 5466 6822 0.51
Best 5466 6822 0.63
CN0002
CN0007
CN0005
CN0001
CN0003
CN0006
CN0004

Figure 16

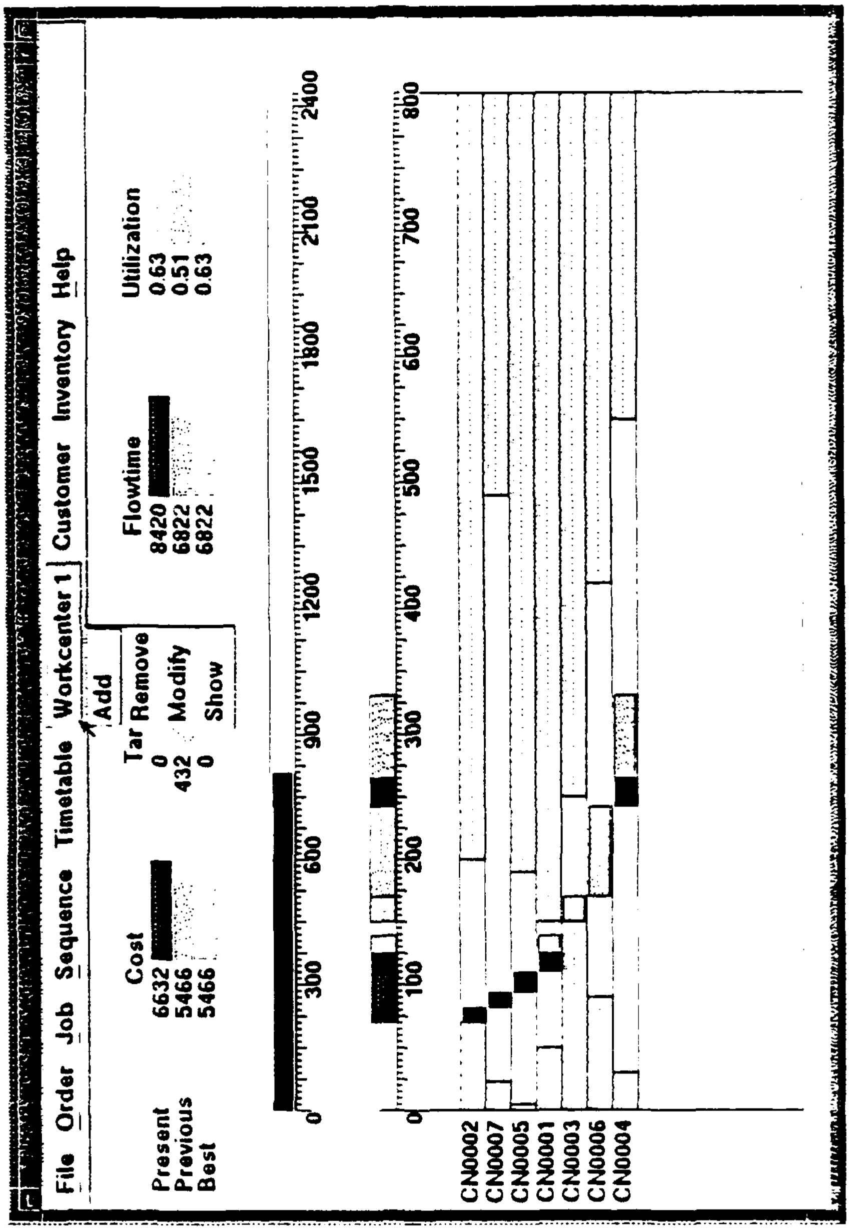
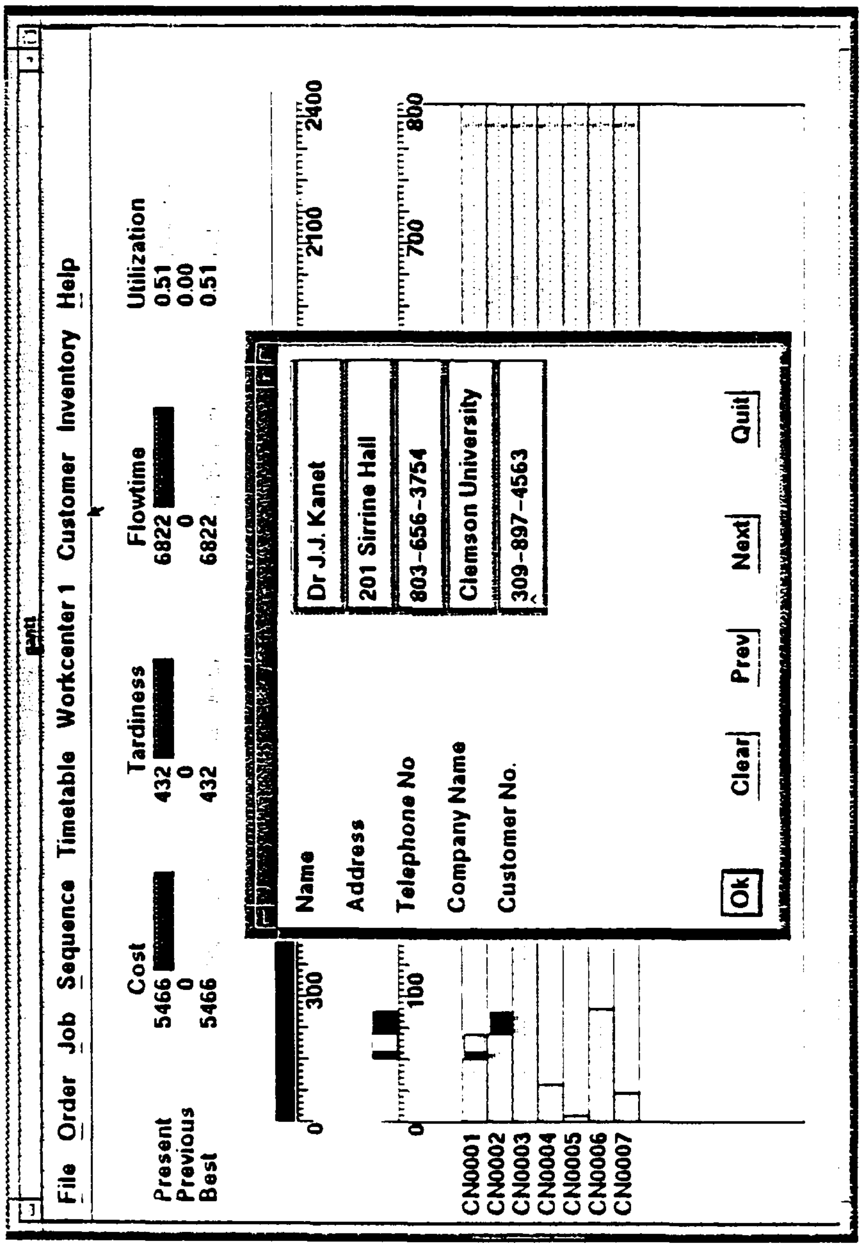

File Order Job Sequence Timetable Workcenter 1 Customer Inventory Help
Add
Remove
Modify
Show
Present
Previous
Best
Cost
6632
5466
5466
Tar
0
432
0
Flowtime
8420
6822
6822
Utilization
0.63
0.51
0.63
0 300 600 900 1200 1500 1800 2100 2400
0 100 200 300 400 500 600 700 800
CN0002
CN0007
CN0005
CN0001
CN0003
CN0006
CN0004

Figure 17



File Order Job Sequence Timetable Workcenter 1 Customer Inventory Help
Present
Previous
Best
Cost
5466
0
5466
Tardiness
432
0
432
Flowtime
6822
0
6822
Utilization
0.51
0.00
0.51
Name
Dr J.J. Kanet
Address
201 Sirrine Hall
Telephone No
803-656-3754
Company Name
Clemson University
Customer No.
309-897-4563
Ok
Clear
Prev
Next
Quit
CN0001
CN0002
CN0003
CN0004
CN0005
CN0006
CN0007

Figure 18

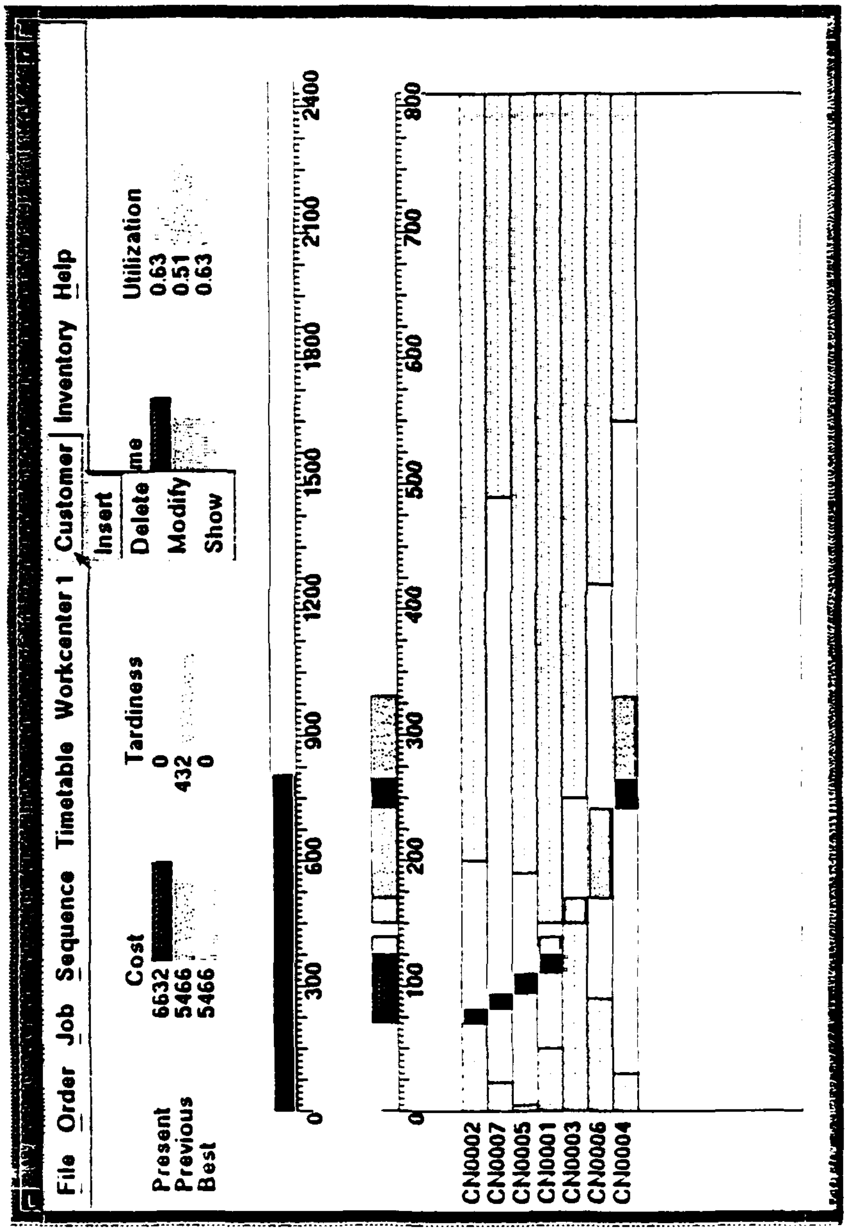

File
Order
Job
Sequence
Timetable
Workcenter 1
Customer
Inventory
Help
Insert
Delete
Modify
Show
Present
Previous
Best
Cost
6632
5466
5466
Tardiness
0
432
0
Utilization
0.63
0.51
0.63
0
300
600
900
1200
1500
1800
2100
2400
0
100
200
300
400
500
600
700
800
CN0002
CN0007
CN0005
CN0001
CN0003
CN0006
CN0004

Figure 19

Show All: The same template shown in Fig 10 appears. The Prev and Next buttons can be used to go through the records available in the database. The fields cannot be changed. (Implementation note: This is done by setting the XmNedit argument of a text widget to False.)

Inventory

This menu shows the inventory level for the present schedule. It operates in a toggle mode. The inventory is shown in place of the 2D Gantt chart. Selecting this menu choice again would show the 2D Gantt chart again. The line in the middle shows the average inventory. Inventory increases at the start time of a job and decreases when a job leaves the workcenter, which by assumption would be the due date if the job is completed earlv or the completion date if the job finishes on time or late. A sample inventory chart is shown in Fig. 20.

IMPLEMENTATION OF QRP INTERFACE USING WIDGETS

The QRP utilizes the rich set of Widgets provided by Motif. Widgets are user interface features that could be used with little or no modification in developing QRP. The ones found useful while developing the software are the MainWindow, BulletinBoard, FileSelection Box, PushButton, Text, Label, Shell, Dialogue, Information, and all the Menuwriting widgets. The introductory screen (Fig. 9) consists of a Shell widget controlling a Main Window widget which controls the DrawingArea widget and the MenuBar widget. The DrawingArea widget controls the Label widget which is the text shown on the screen. The MenuBar widget in turn controls the Cascade widget which controls the Pulldownmenu widget which controls the Pushbutton widgets. The menu callbacks are implemented as callbacks to these Pushbutton widgets. The Menu widget structure is a fixed structure to be followed for showing menus on the screen.

The template which appears when an Order, Workcenter, or Customer submenu is selected has the hierarchy of Shell widget controlling a BulletinBoard widget. The BulletinBoard widget controls the Label widget which displays the labels on the left hand side of each template and the Frame widgets which control the Form widgets which in turn control the Text widget. The Frame widget is necessary to give the outline and the Form widget is necessary to control the Text widget to maintain the relative positions of the text entry field on the template. The Text widget is the widget that takes input from the keyboard and outputs it on the screen. The information (entered text) is then obtained by making calls to the Text widget.

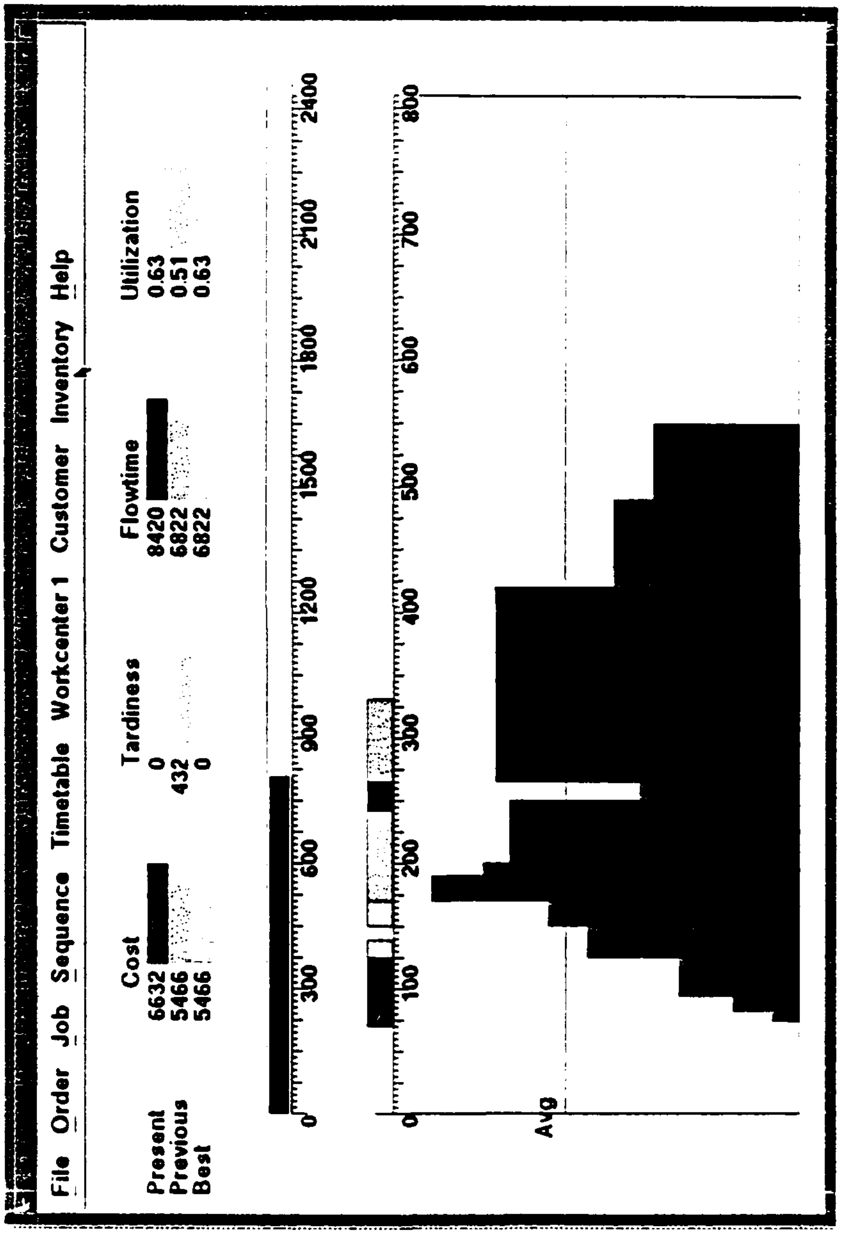

File Order Job Sequence Timetable Workcenter 1 Customer Inventory Help
Cost
Tardiness
Flowtime
Utilization
Present 6632 0 8420 0.63
Previous 5466 432 6822 0.51
Best 5466 0 6822 0.63
0 300 600 900 1200 1500 1800 2100 2400
0 100 200 300 400 500 600 700 800
Avg

Figure 20

IV. NETWORKING ALGORITHM DEVELOPMENT

The purpose of this part of the project was to investigate the employment of cooperating, communicating QRP systems at several workcenters in an organization. In a simple but important case, the workcenters would be associated in a supplier/consumer relationship, with two or more related in a chain. To make a network of this type useful requires addressing some difficult issues. One of the most important requirements of such a network is to somehow maintain integrity among the various schedules. For example, assume that the output of workcenter A is used by workcenter B to manufacture a final product. It is not satisfactory for the schedule for workcenter B to indicate that the input for a job will be ready earlier than that input is scheduled to be provided by workcenter A. Yet, if the network is not controlled carefully, and if schedule changes are allowed to be made in isolation, without coordinating with other workcenters, integrity will likely be violated.

In our work on networking algorithms, we considered a restricted but commonly occurring situation in which managers of the various workcenters have already selected schedules, but for some reason need to make changes to the schedules. Reasons could include a change in customer requirements or an opportunity to implement a lower cost schedule. We developed a set of algorithms which can be used to change the schedule at a workcenter in such a manner that the revised schedule will be consistent with the schedules of all other workcenters in the network. The algorithms provide the following operations to the manager: Insert, Delete, Modify Ask (request of a modification), and Information Message (to notify other workcenters of a change). Also, the algorithms include messages which can be generated automatically by software to help implement the aforementioned user operations: Modify Force, Modify OK, Modify Reject, and Modify Update (these are described below).

The remainder of this section provides a description of the networking algorithms. The Insert and Modify Force algorithms have been implemented and demonstrated on networked computers. All the algorithms have been tested by "structured walkthroughs" involving members of the QRP team. They appear to be sound, and are ready to convert to computer code and test.

ALGORITHMS FOR HANDLING MESSAGE TYPES

1) Insert

Brief Description:

An order can be inserted on any workcenter. There would be three passes before the system would reach steady state. In the first stage the order in the Insert message just passes down until it reaches those workcenters which do not have any successors where it is scheduled first. These workcenters then send a modify force message to their parent informing them of their completion date which would become the ready date of the parent. Then when the completion date is obtained from the successor workcenter the parent workcenter schedules the order and sends its start date to all its children. The start date of the parent would be the actual due date of the children. This set of messages goes on until it is finally scheduled on the workcenter where it had originated. Then the steady state is reached.

Algorithm Used When an Insert Message is Received:

```
Insert the order in the order database.
If the current workcenter has successor workcenters
     send an Insert message to all its children.
Else{
     Schedule the order;
     If current workcenter is not the originator
     send a Modify Force message to its parent.
    }
```

Message Body:

Message Type
Sending Workcenter
Order No.
Process Type
Data = Due Date
Early Penalty
Late Penalty

2) Delete

Generated by:

This message type is initially generated by the user when he wants to delete an order. It is then generated by successive workcenters to propagate this information.

Comments:

The order is deleted from the top level workcenter down to the lowermost level workcenter where it was scheduled.

Algorithm Used When a Delete Message is Received:

```
Delete the order entry from the current workcenters order
database.
If current workcenter has successor workcenter
     send a Delete message to all the successor
     workcenters.
```

Message Body:

Message Type
Sending Workcenter
Order No.

3) Modify Force

Generating by:

Modify Force message is generated automatically when an Insert message is generated. It can also be generated by the user to effect an important change without asking its predecessors.

Limitations:

No Modify Force messages allowed from parents. Because that might force the order scheduled on the lowermost level workcenter to move beyond the ready date.

Algorithm Used When a Modify Force Message is Received:

```
In the receiver workcenter's order entry mark the bit
saying this child's operation is scheduled.
If the completion date in the message is greater than the
ready date
      ready date = completion date in the message.
If all the child operations are complete
     {
      Schedule the order.
      Send an Information message [order no, time = start
      time,] to all its children informing the new
      completion date;
      Send a Modify Force [order no, date = completion
      time, date,] message to its parent.
     }
```

Message Body:

Message Type
Sending Workcenter
Order No.
Data

4) Modify Ask

Generated by:

This message type is always generated initially by the user either when he wants to shift an order to the left or the right of the present completion date. This message can only be generated after steady state has been reached.

Comments:

This is a request, and depending on the answer from the child or parent it can succeed or fail. It can come either from a child or a parent. If the user wants to move the order to the left then messages are sent to the children and if he wants to move to the right then a Modify Ask message is sent to the parent.

Algorithm Used When a Modify Ask Message is Received:

```
If the schedule cannot be modified as requested
      send a Modify Reject message.

Else
     {
      Put the order on hold.
      If the message is from a child
          {
           Reserve a space on the left.
           If current workcenter has a parent
               send a Modify Ask message to its parent.
           Else
               send a Modify OK message to the sender.
          }
      Else
          {
           Reserve a space on the right.
           If current workcenter and operation has
           children
               send a Modify Ask to all its children.
           Else
               send a Modify OK to the sender.
          }
     }
```

Message Body:

Message Type
Sending Workcenter
Order No.
Data = start date/completion date

5) Modify OK

Generated by:

This message is always generated in response to a Modify Ask message.

Algorithm Used When a Modify OK Message is Received:

```
If the message is from the parent and receiver is not the
originator
      {
       Send a Modify OK to the originator child.
       Quit.
      }
If the message is from the parent and receiver is the
originator
       Send a Modify Update to the parent.
If the Modify OK is from the child and receiver is
not the originator
      {
       Update the Modify OK bit for that child.
       If all the children have sent a Modify OK bit
       Send a Modify OK to the parent.
       Quit.
      }
If the Modify OK is from the child and receiver is the
originator
      {
       Update the Modify OK bit for that child.
       If all the children have not sent a Modify OK
       Quit.
      }
Else
      {
       Send a Modify Update to its children.
       Remove the order from hold.
       Put it in the reserved space.
       Free the previous space taken up by the order
       Quit.
      }
```

Message Body:

Message Type
Sending Workcenter
Order No.

6) Modify Reject

Generated by:

This message type is always generated in response to a Modify Ask message.

Algorithm Used When a Modify Reject Message is Received:

If message is from the child and current workcenter not the originator
send a Modify Reject to the parent.
Else if message is from the parent and current workcenter is not the originator
send a Modify Reject to the child who asked.
Free up the reserved space.
Remove the order from hold.

Message Body:
Message Type
Sending Workcenter
Order No.

7) Modify Update

Generated by:

Generated only when the originator parent of a Modify Ask has received a Modify OK from all its children or the originator child has received a Modify OK from the parent.

Algorithm Used When a Modify Update Message is Received:

Put the order in the reserved space.
Release the previous space occupied by the order.
Remove the order from hold.
If the sender is a child and a parent is present
send a Modify Update to the parent.
Else if the sender is a parent and a child is present
Send a Modify Update to all the children.

Message Body:

Message Type
Sending Workcenter
Order No.

8) Information Message

Generated by:

Generated when the user tries to shift a job either to the left or right after the steady state has been achieved and the movement is in the feasible region (i.e., between the ready date and the due date). If the job is moved to the right then a message is sent to the parent to update its ready date if appropriate. If the job is moved to the left then a message is sent to all the children to tell them of their new due date.

Algorithm Used When an Information Message is Received:

If the message is from the parent
    Update the due date.
Else
    If the completion date in the message is greater than the ready date
    Update the ready date.

Message Body:

Message Type
Sending Workcenter
Order No.
Data = start date / completion date

<u>Advantages:</u>

1) Modify Ask gives more leverage to the user.
2) Modify Force can take care of unforeseen circumstances.
3) Modify Ask needs a lot of message passing.
4) Modify Force is easier to implement.

<u>Schedule Algorithm Suggested</u>

Schedule ASAP after the ready date.

V. BIBLIOGRAPHY

Adelsberger, H.H., and J. J. Kanet, "The Leitstand--A New Tool in Computer-Aided Manufacturing Scheduling," Proceedings of the 3rd ORSA/TIMS Conference on Flexible Manufacturing Systems, (Stecke and Suri, ed.), Elsevier, Amsterdam, 1989, pp. 231-236.

Adelsberger, H.H., and J. J. Kanet, "The Leitstand--A New Tool for Computer-Integrated Manufacturing," Production and Inventory Management, Vol. 32, No.1, (1st Quarter, 1991), pp. 43-48.

Abraham, C., Dietrich, B., Graves, S., Maxwell, W., and C. Yano, "A Research Agenda for Models to Plan and Schedule Manufacturing Systems," report from NSF workshop "Scheduling the Factory of the Future: A Research Planning Session," Univ. of Penn., March, 1985.

Bagchi, U., Sullivan, R. S. and Y. L. Chang, "Minimizing Mean Absolute Deviation of Completion Times About a Common Due Date," Naval Res. Logist. Quart., 33, 1986, pp. 227-240.

Bagchi, U., "Simultaneous Minimization of Mean and Variance of Flow Time and Waiting Time," Working paper. The University of Texas, 1987.

Baker, K. R., Introduction to Sequencing and Scheduling, John Wiley & Sons, New York, 1974.

Baker, K. R., and J. B. Martin, "An Experimental Comparison of Solution Algorithms for the Single-Machine Tardiness Problem," Naval Res. Logist. Quart., Vol. 21, No. 1, (March 1974), pp. 187-199.

Baker, K. R., "Sequencing Rules and Due-Date Assignments in a Job Shop," Management Science, Vol. 30, No. 9 (Sept., 1984a), pp. 1093-1104.

Baker, K. R., "The Effects of Input Control on the Performance of a Simple Scheduling Model," Journal Operations Management, Vol. 4, No. 1, (Jan-Mar, 1984b), pp. 99-112.

Baker, K. R., and J. J. Kanet, "Job Shop Scheduling with Modified Due Dates," Journal of Operations Management, Vol. 4, No. 1 (Nov., 1983), pp. 11-22.

Baker, K. R., and G. D. Scudder, "Sequencing with Earliness and Tardiness Penalties: A Review," Working Paper No. 226. Dartmouth College, 1988.

Bechte, W., "Theory and Practice of Load-Oriented Manufacturing Control," International Journal of Production Research, Vol. 26, No. 3, (March, 1988), pp. 375-395.

Bertrand, J. W. M., "The Use of Workload Information to Control Job Lateness in Controlled and Uncontrolled Release Production Systems," Journal of Operations Management, Vol. 3, No. 2 (Feb., 1983), pp.79-92.

Bitran, G. R., Haas, E. A., and A. C. Hax, "Hierarchical Production Planning: A Two-Staged System," Operations Research, Vol. 30, No. 2 (Mar.-Apr., 1982).

Bitran, G. R., and A. C. Hax, "On The Design of Hierarchical Production Planning Systems," Decision Sciences, Vol. 8, No. 1 (Jan., 1977), pp. 28-54.

Blackstone, J. H. Jr., Phillips, D. T., and G. L. Hogg, "A State-of-the-Art Survey of Dispatching Rules for Manufacturing Job Shop Operations," International Journal of Production Research, Vol. 20, No. 1 (Jan./Feb., 1982), pp. 27-45.

Bobrowski, P. M., and Park, P. S., "Work Release Strategies in a Dual Resource Constrained Job Shop," Omega, Vol. 17, No. 2, (Feb., 1989), pp. 177-188.

Callahan, R. L., "MRP is a Charade, It Offers No Solutions," Manufacturing Week, Dec. 7, 1987, p. 15.

Christy, D. P., Job Shop Scheduling with Stochastic Due Dates, unpublished doctoral dissertation, University of Georgia, 1984.

Christy, D. P., and J. J. Kanet, "Open Order Rescheduling in Job Shops with Demand Uncertainty: A Simulation Study," Decision Sciences, Vol. 19, No. 4, (Fall, 1988), pp. 801-818.

Coffman, E. G., Computer and Job Shop Scheduling Theory, John Wiley and Sons, New York, 1976.

Cohen, S., and J. Zysman, Manufacturing Matters: The Myth of a Post-Industrial Economy, Basic Books, New York, 1987.

Conway, R. W., and W. L. Maxwell, "Low-Level Interactive Scheduling," National Bureau of Standards Symposium on Real Time Optimization for Automated Manufacturing Facilities, April 1, 1986.

Conway, R. W., Maxwell, W. L., and L. W. Miller, Theory of Scheduling, Addison-Wesley, Reading, Mass., 1967.

Davis, J. S., and J. J. Kanet, "Single Machine Scheduling with Non-Regular Convex Completion Costs," (Under Review, 1991).

Eilon, S., and I.G. Chowdhury, "Minimizing Waiting Time Variance in the Single Machine Problem," Mgmt. Sci., 23, (1977), pp. 567-575.

Faaland, B., and T. Schmitt, "Scheduling Tasks with Due Dates in a Fabrication/Assembly Process, " Opns. Res. 35 (1987), pp. 378-388.

Fox, M.S., and S.F. Smith, "ISIS -- A Knowledge-Based System for Factory Scheduling," Expert Systems, Vol.1, No.1 (1st quarter, 1984), pp. 25-49.

French, S., Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop. Halstead Press, New York, 1982.

Fry, T. D., Armstrong, R. D., and J. H. Blackstone, "Minimizing Weighted Absolute Deviation in Single Machine Scheduling," IEE Transactions, Vol. 19, No. 4 (1987), pp. 445-450.

Gershwin, S. B., Hildebrant, R. R., Suri, R., and S. K. Mitter, "A Control Theorist's Perspective on Recent Trends in Manufacturing Systems," report no. LIDS-R-1408, MIT, revision of March, 1985.

Giffler, B., and G. L. Thompson, "Algorithms for Solving Production Scheduling Problems," Operations Research, Vol. 8, No. 4 (July, 1960).

Graves, S.C., "A Review of Production Scheduling," Operations Research, Vol. 29, No. 4, (Apr., 1981), pp. 646-675.

Hall, N. G., "Single and Multiple-Processor Models for Minimizing Completion Time Variance," Naval Res. Logist. Quart. Vol. 33 (1986), pp. 49-54.

Hax, A. C., and D. Candea, Production and Inventory Management, Prentice-Hall, Inc., Englewood Cliffs, 1984.

Hayes, R. H., and S. C. Wheelwright, Restoring Our Competitive Edge, John Wiley & Sons, New York, 1984.

Haupt, R., "A Survey of Priority Rule-Based Scheduling," OR Spectrum, Vol. 11, No. 3, (Mar., 1989), pp. 3-16.

"IBM Reigns in Dynamic MRP II Marketplace," Computerworld, Jan. 27, 1986, p. 37.

Irastorza, J.C., and Deane, R. H., "A Loading and Balancing Methodology for Job Shop Control," AIIE Transactions, Vol. 6, No. 4, (Oct.-Dec., 1974), pp. 302-307.

Jones, C.V., and W.L. Maxwell, "A System for Manufacturing Scheduling with Interactive Computer Graphics," IIE Transactions, Vol. 18, No. 3 (Sept., 1986), pp. 298-303.

Kanet, J. J., "Job Shop Scheduling to Meet Due Dates: A Simulation Study," unpublished doctoral dissertation, The Pennsylvania State University, 1979.

Kanet, J. J., "Minimizing the Average Deviation of Job Completion Times About a Common Due Date," Naval Res. Logist. Quart., Vol. 28 (1981), pp. 643-651.

Kanet, J. J., "Minimizing Variation of Flow Time in Single Machine Systems," Mgmt. Sci. Vol. 27 (1981), pp. 1453-1459.

Kanet, J. J., "Toward Understanding Lead Times in MRP Systems," Production & Inventory Management, Vol.23, No.3, (3rd quarter, 1982), pp. 1-14.

Kanet, J. J., "Inventory Planning and Control at Black & Decker", Production & Inventory Management, Vol. 25, No. 3, (3rd quarter, 1984), pp. 9-22.

Kanet, J. J., "Microcomputer Aided Capacity Planning Using Animated Color Graphics," American Production and Inventory Control Society 1986 Fall Seminar Proceedings, Orlando, Fla., Sept. 8-10, 1986a.

Kanet, J. J., "Tactically Delayed Versus Non-Delay Scheduling: An Experimental Investigation," European Journal of Operational Research, Vol. 24, No. 1 (Jan., 1986b), pp. 99-105.

Kanet, J. J., "Measuring Manufacturing Performance: Implications for Research and Practice," working paper, Clemson University, 1987.

Kanet, J. J., "Load-Limited Order Release in Job Shop Scheduling Systems", Journal of Operations Management, Vol. 7, No. 3, (Dec., 1988a), pp. 44-58.

Kanet, J. J., "MRP-96: Time to Rethink Manufacturing Logistics", Production & Inventory Management, Vol. 29, No. 2 (2nd Qtr., 1988b), pp. 57-61.

Kanet, J. J., and H. H. Adelsberger, "Expert Systems in Production Scheduling," European Journal of Operational Research, Vol. 29, No. 1 (April, 1987), pp. 51-59.

Kanet, J.J., and D. P. Christy, "Manufacturing Systems with Forbidden Early Order Departure," International Journal of Production Research, Vol. 22, No. 1 (1984), pp. 41-50.

Kanet, J. J., and R. Dattero, "An Alternative Approach to Manufacturing Logistics," Working Paper, Department of Business Analysis and Research, Texas A&M University, 1986.

Kanet, J.J., and J. C. Hayya, "Priority Dispatching with Operation Due Dates in a Job Shop," Journal of Operations Management, Vol. 2, No. 3 (May, 1982), pp.167-176.

Kanet, J. J., and R. A. Jacobs, "Decision-Making in Production Management Using Cash Flow Analysis," Working Paper 87-1, Department of Management, Clemson University, 1987.

Kanet, J. J., and V Sridharan, "Production Scheduling with an Electronic Leitstand: Toward a New Architecture for Manufacturing Planning", (Baker, J.R., and Sumichrast, R.T., ed), Proceedings of the 25th Annual Meeting of The Institute of Management Sciences, South Eastern Chapter, October 5-6, 1989, Myrtle Beach, SC, pp. 376-378.

Kanet, J. J., and V Sridharan, "The Electronic Leitstand: A New Tool for Shop Scheduling," Manufacturing Review, Vol. 3, No. 3 (September, 1990), pp. 161-170.

Karmarkar, U. S., "Lot Sizes, Lead Times, and In-Process Inventories," Management Science, Vol. 33, No. 4, (Mar., 1987), pp. 409-418).

Karmarkar, U. S., "Capacity Loading and Release Planning with Work-in-Process and Leadtimes," Journal of Manufacturing and Operations Management, Vol. 2, No. 2, 1989a, pp. 105-123.

Karmarkar, U. S., "Getting Control of Just-In-Time," Harvard Business Review, Sept.-Oct., 1989b, pp. 358-367.

Kurbel, K., and Meynert, J., "Flexibilität in der Fertigunsteuerung durch einen Elektronischen Leitstand," working paper, Universität Dortmund, June, 1988.

Law, A.V., and Kelton, W.D., Simulation Modeling and Analysis, McGraw-Hill, New York, 1982.

Lenstra, J. K., Rinnoy Kan, A. H. G., and P. Brucker, "Complexity of Machine Scheduling Problems," Annals of Discrete Mathematics, Vol. 1 (1977), pp. 343-362.

Lippe, J. von, and Dengler, U., JOBPLAN User's Manual for Demonstration Version, Siemens AG, Munich, West Germany, May, 1988.

Mather, H. "MRP II Won't Schedule the Factory of the Future," CIM Review, Vol. 3, No. 1 (Fall, 1986), pp. 64-68.

Maxwell, W. L., Muckstadt, J. A., Thomas, L. J., and J. VanderEecken, "A Modeling Framework for Planning and Control of Production in Discrete Parts Manufacturing and Assembly Systems," Interfaces, Vol. 13, No. 6, Dec., 1983, pp. 92-104.

Maxwell, W. L., and J. A. Muckstadt, "Variance Minimization in Single Machine Sequencing Problems," Mgmt. Sci., Vol. 18 (1985), pp. 518-528.

McClain, J. O., Maxwell W. L., Muckstadt, J. A., Thomas, L. J., and E. L. Weiss, "On MRP Lot-Sizing," Management Science, Vol. 28, No. 5, (May 1982).

Melnyk., S. A., Carter, P. L., Dilts, D. M., and D. M. Lyth, Shop Floor Control, Dow Jones-Irwin, Homewood, IL, 1985.

Melnyk, S. A., and P. L. Carter, Production Activity Control: Practical Guide, Dow Jones-Irwin, Homewood, IL, 1987.

Melnyk, S. A., and G. L. Ragatz, "Order Review/Release: Research Issues and Perspectives," International Journal of Production Research, Vol. 27, No. 7, (July, 1989), pp. 1081-1096.

Merten, A. G., and M. E. Muller, "Variance Minimization in Single Machine Sequencing Problems," Mgmt. Sci., Vol. 18, (1972), pp. 518-528.

Mertens, P., and J. J. Kanet, "Expert Systems in Production Management: An Assessment," Journal of Operations Management, Vol. 6, No. 4 (Aug., 1986), pp. 393-404.

Monden, Y., Toyota Production System, Industrial Engineering and Management Press, Institute of Industrial Engineers, Norcross, Ga., 1983.

Morton, T. E., Lawrence, S. R., and G. L. Thompson, "MRP-Star Patriarch's Planning Module," partial draft working paper, Carnegie- Mellon University, June 5, 1986.

Morton, T.E., S.R. Lawrence, S. Rajagopalan, and S. Kekre, "SCHED-STAR: A Price-Based Shop Scheduling Module," Journal of Manufacturing and Operations Management, Vol. 1, No. 2 (1988), pp. 131-181.

Neter, J. and W. Wasserman, Applied Linear Statistical Models, Richard D. Irwin, Homewood, IL, 1974.

Orlicky, Joseph A., Material Requirements Planning, McGraw-Hill, New York, 1975.

Ow, P. S., and T. E. Morton, "Filtered Beam Search in Scheduling," International Journal of Production Research, Vol. 26, No. 1 (1988), pp. 35-62.

Ow, P. S., and T. E. Morton, "The Single Machine Early/Tardy Problem," Mgmt. Sci., Vol. 35, No. 2 (February 1989), pp. 177-191.

Pearl, J., Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley, Reading, Mass., 1984.

Rappaport, A., Creating Shareholder Value, The Free Press, New York, 1986.

Rinnoy Kan, A. H. G., Machine Scheduling Problems: Classification, Complexity and Computations. Martinus Nijhoff, The Hague, 1976.

Roundy, R., "Production Planning and Scheduling," Manufacturing Processes, Machines, and Systems, 13th Conference on Production Research and Technology, NSF, 1986, pp. 385-392.

Schonberger, R. S., World Class Manufacturing, The Free Press, New York, 1986.

Schrage, L., "Minimizing the Time-in System Variance for a Finite Job-Set," Mgmt. Sci., Vol. 21 (1975), pp. 540-543.

Shimoyashira, S., Isoda, K., and H. Awane, "Input Scheduling and Load Balance Control for a Job Shop," International Journal of Production Research, Vol. 22, No. 4, (Aug.-Sept., 1984), pp. 597-605.

Sidney, J. B., "Single-Machine Scheduling with Earliness and Tardiness Penalties," Opns. Res., Vol. 25 (1977), pp. 62-69.

Smith, W. E., "Various Optimizers for Single Stage Production," Naval Research Logistics Quarterly, Vol. 3, No. 1 (Mar., 1956), pp. 59-66.

Srinivasan, V., "A Hybrid Algorithm for the One Machine Sequencing Problem to Minimize Total Tardiness," Naval Res. Logist. Quart., Vol. 18, (September 1971), pp. 317-327.

Sundaraghaven, P. S., and M. U. Ahmed, "Minimizing the Sum of Absolute Lateness in Single-Machine and Multi-Machine Scheduling," Naval Res. Logist. Quart. Vol. 31 (1984), pp. 325-333.

Vani, V., and M. Raghavachari, "Deterministic and Random Single Machine Sequencing with Variance Minimization," Opns. Res., Vol. 35 (1987), pp. 111-120.

Vollmann, T.E., Berry, W.L., and D.C. Whybark, Manufacturing Planning and Control Systems, Richard D. Irwin, Inc., 1984.

VI. TECHNOLOGY TRANSFER

An important aspect of this project has been the transfer of the knowledge gained so that others may build on it. To this end, we published the results of our evaluation at both research and apparel trade conferences, and demonstrated the algorithm at the Apparel Advanced Manufacturing Technology Demonstration (AAMTD) site at Pendleton, South Carolina. Efforts in technology transfer occurred in the following major categories:

Conference and Symposia Participation
Technical Publications
Technical Meetings

CONFERENCE AND SYMPOSIA PARTICIPATION

August, 1989

On August 14-16, 1989, Kanet presented the paper "The Leitstand -- A New Tool in Computer-Aided Manufacturing Scheduling" (with H. H. Adelsberger, Technical University of Denmark) at the Third ORSA/TIMS Special Interest Conference on Flexible Manufacturing Systems: Operations Research Models and Applications, held at Massachusetts Institute of Technology, Cambridge, Mass.

October, 1989

Davis participated in the conference on Object-Oriented Systems and Languages on October 1-4, 1989, at New Orleans, LA, helping to bring the Quick Response project team up to the state of the art in developing systems with the object-oriented approach. Also, he got information about object-oriented languages, in particular C++ which was evaluated for use in this project. Benefits of the object-oriented approach include easier debugging and more straightforward extension and modification of the system.

On October 16-18, 1989, Kanet participated in the ORSA/TIMS joint national meeting in New York. There he organized the paper session "New Developments in Manufacturing Planning and Scheduling" and presented the paper "The Leitstand: A New Tool in Computer Aided Manufacturing Scheduling (with H. H. Adelsberger, TU Denmark). At this conference Davis and Kanet presented the paper "Quick Response Planning and Scheduling."

On October 27, 1989, Kanet served as a panel member of a special presentation "Manufacturing Systems for the 1990's: North America, Europe, Japan," at the American Production and Inventory Control Society's 32nd International Conference, Orlando, FL.

In October 1989, the Quick Response team presented a briefing and demonstration at the annual contract briefing at Clemson Apparel Research. Michael O'Rourke, an undergraduate student who worked on the project in the summer of 1989, presented results of his work in investigating the applicability of neural networks to job scheduling. O'Rourke continued to investigate this problem with funding support from a National Science Foundation program administered by Prof. Davis.

February, 1990

Davis and Kanet traveled to Philadelphia, Pennsylvania for the Advanced Apparel Manufacturing Technology Demonstration on February 14-16, 1990. There they presented a summary of their work. While at the conference, they toured the Defense Personnel Support Center.

April, 1990

During the month of April the research team was involved in the Automated Manufacturing (AM 90) Exposition in Greenville, SC, which was held at the Palmetto International Center on April 3-5. Clemson had a boot set-up with Batelle Institute for demonstrating the joint venture between Batelle and Clemson University. The very early prototype of the Quick Response Planner as well as the Siemens Leitstand which was donated to Clemson by Siemens, Germany, were demonstrated.

May, 1990

On May 14-16, 1990, Kanet participated in the Fourth International Conference on "Expert Systems in Production and Operations Management" at Hilton Head, SC. There he organized the panel session entitled "Intelligent Shop Scheduling and Control" with participation from Professor Heimo H. Adelsberger, Technical University of Denmark, Mr. Hermann Havermann, AHP Havermann & Partner GmbH, Munich, Germany, Mr. Udo Dengler, Siemens AG Research Lab, Munich, Germany, Mr. Dieter Steinmann, University of Saarland-Saarbrücken, and Dr. Jack C. Peck, Computer Science Department of Clemson University. Kanet also participated in a session entitled "Manufacturing Planning and Control Systems: Past, Present and Future" with Mr. Tom Reif, Ingersoll Engineering, and Mr. Josef Schengili, Numetrix, Brussels, Belgium.

June, 1990

On June 28, 1990, Kanet participated in the 12th Triennial Conference on Operations Research on "Wisdom for the Problems of Today" of the International Federation of Operational Research Societies in Athens, Greece. There Mr. Joe von Lippe, Siemens AG, Munich, Germany, and Kanet presented the paper "Applying the Jobplan Leitstand: Experiences and Projections." Jobplan is a prototype leitstand capable of planning and coordinating up to 20

production resources. Jobplan has been donated to Clemson University researchers to aid in the development of scheduling projects such as the Quick Response Planner.

August, 1990

On August 28, 1990, Kanet participated in a meeting held at Cameron Station in Alexandria, VA, for all researchers involved in the DLA sponsored production planning research projects. There Kanet shared experiences with other researchers from Georgia Tech, FIT, and Clemson University concerning production scheduling issues for apparel manufacturers.

October, 1990

On October 4, 1990, Kanet presented a one-day seminar on Production and Inventory Planning entitled "Production Scheduling for Quick Response: The New Way of Life for the U.S. Apparel Industry" at the Clemson Apparel Research Center in Pendleton, S.C. Attendees were individuals from the apparel industry who are responsible for creating production plans and who needed to become familiar with the latest developments in computer-aided production scheduling techniques. The purpose of the seminar was to provide the participants a new perspective on how proper attention to scheduling yields a major competitive advantage. Attendees got to see first hand some of the latest computer-based tools for production scheduling and control.

On October 5, 1990, Kanet presented a paper entitled "Intelligent Search in Production Scheduling" (Co-author V Sridharan) at the 26th Annual Meeting of the Southeastern Chapter of The Institute of Management Sciences in Myrtle Beach, SC.

Davis travelled to Washington, D.C., on October 15-17, 1990 to participate in the First Annual Production and Operations Management Systems Conference. He presented the paper "Single Machine Scheduling with Convex Non-Regular Completion Costs" and chaired a session on single-machine scheduling. This presentation included a summary of the Quick Response Planner project. Several in the audience expressed interest in any insight we had gained on the separability of sequencing and timetabling in the search for an optimal schedule. The consensus was that such a separability is beyond the state of the art, and it might be theoretically impossible to achieve separation in any useful way in an optimal algorithm. Opinions of the conference attendees supported the approach we used in the Quick Response Planner project. We allowed manual separation of sequencing and timetabling in searching for a near-optimal solution.

November, 1990

On November 19-21, 1990, Kanet attended the 21st Annual Decision Sciences Institute meeting in San Diego, California. There he participated along with V Sridharan (panel chairman) in a panel discussion entitled "Order Review - Release: A Mirage?" with professors from San Jose State, Michigan State, Texas Christian University, and the University of Hannover.

February, 1991

On February 11-12, 1991 Davis and Kanet attended the Apparel Research Conference, sponsored by the Defense Logistic Agency, at Clemson, SC. There they provided a briefing on the status of the DLA-sponsored project "Production Planning for Quick Response."

May, 1991

On May 13-15, 1991, Kanet travelled to Nashville, Tennessee, to participate in the TIMS/ORSA joint national meeting. There he organized the paper session entitled "Real-Time Production Scheduling Systems" which included the following papers:

- "A Decision Support System for Interactive Production Scheduling & Control: The Knowledge-Based Leitstand KBL," H. H. Adelsberger, TU Denmark;
- "A Decision Making Approach to Finite Production Scheduling," G. Chryssolouris, MIT;
- "Facts and Results of an Electronic Gantt-Chart Tool: The Leitstand," J. V. Lippe, Siemens Ag, Munich;
- "A Data Model for Decentralized Real-Time Scheduling," A.-W Scheer, Saarbrücken, Germany.

TECHNICAL PUBLICATIONS

Adelsberger, H. H., and J. J. Kanet, "The Leitstand - A New Tool in Computer-Aided Manufacturing Scheduling," Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems, (Stecke and Suri, ed.) Elsevier, Amsterdam, 1989, pp. 231-236.

Adelsberger, H. H. and J. J. Kanet, "The Leitstand -- A New Tool for Computer-Integrated Manufacturing," Production and Inventory Management, (Vol. 32, No. 1, 1st Quarter, 1991), pp. 43-48.

Davis, J. S., and J. J. Kanet, "Single Machine Scheduling with Non-Regular Convex Completion Costs," (Under Review).

Kanet, J. J. and V Sridharan, "Resource Scheduling Using an Electronic Leitstand: An Assessment," Proceedings of the SETIMS Annual Conference, Myrtle Beach, Oct 5-7, 1989, pp. 376-378.

Kanet, J. J., "Measuring Manufacturing Performance: Implications for Research and Practice," Proceedings of the 5th International Conference of the Operations Management Association UK, Warwick, June 26-27, 1990, pp. 318-336.

Kanet, J. J. and V Sridharan, "The Leitstand: A New Tool for Shop Scheduling." Manufacturing Review, Vol. 3, No. 3 (September, 1990), pp. 161-170.

Kanet, J. J., "Anwendungssoftware für die Produktion in den USA -- aktueller Stand und Perspektiven für die 90er Jahre," Proceedings of the Fachtagung 'Anwendungssoftware der 90er Jahre,' sponsored by the University of the Saarland and the Ausschuß für Fertigung e. V., FRG, April 25-26, 1991.

Kanet, J. J., "The Leitstand: Real Decision Support for Production Scheduling and Control," (To appear: Production and Inventory Management Review).

TECHNICAL MEETINGS

October, 1989

On October 12, 1989, Kanet visited the Manufacturing Systems Research Group of Philips Laboratories, Briarcliff, NY, where he provided a presentation of Clemson University's research in advanced manufacturing planning and control systems.

November, 1989

On November 21, 1989, Mr. Tom Detscher of American Software visited the Clemson Apparel Research facility at Pendleton, SC. Mr. Detscher is the Vice President of the American Production and Inventory Control Society's (APICS) Special Interest Group for Textile and Apparel Industry. APICS is an international society of over 60,000 production and inventory management professionals from all areas of manufacturing. Mr. Detscher is also a Senior Consultant for American Software of Atlanta, GA. American Software is a leading software house in production and inventory planning software. The purpose of his visit was to be introduced to the activities at CAR particularly those relating to production planning and scheduling. During his visit, Mr. Detscher was provided an introduction to the Quick Response Planning project.

December, 1989

On December 8, 1989, representatives from three different sites of Milliken Co. reviewed the research progress in production planning and scheduling. They included Mr. Steve Freudenthal, Project Manager Expert Systems, Spartanburg headquarters; Ms. Jamie Morgan, Production Planner at their Cushman plant in Williamston, SC; Mr.

On May 17, 1990, Professor Heimo H. Adelsberger of the Technical University of Denmark, Ms. Karen Biddle, Mr. Steve Freudenthal, Mr. Bill Edwards, Ms. Mary Alice Cooper, and Mr. Cary Grant of Milliken & Co. in Spartanburg, and Mr. Dieter Steinmann of the University of Saarland, Saarbrücken, Germany, met at the Clemson Apparel Research Center (CAR) to review the various projects which Kanet and Davis have underway.

On May 18, 1990, Mr. Hermann Havermann visited CAR. Mr. Havermann had donated a copy of his leitstand software product which we used for embedding some of our ideas regarding scheduling algorithms into his software, and tested in a reorder application as in Milliken's facilities.

June, 1990

On June 20, 1990, Kanet visited Micro Dynamics Corporation in Dallas, Texas, where he met with Dr. Ron Gordon to discuss possible cooperation with the Quick Response Planner project. Kanet gave a demonstration of the leitstand prototype that AHP Havermann had donated to Clemson University.

July, 1990

In July, 1990, Davis visited Mr. Christian P. Vogt, Vice President for Information Systems at the Medical Center Hospital of Vermont. Mr. Vogt was developing a large scale scheduling system for hospital operations. Davis and Vogt exchanged information about scheduling projects. Vogt's project and ours had a lot in common. For example, the Medical Center must efficiently schedule each critical resource such as an operating room. The room is analogous to a single machine. It processes jobs (operations on patients) and there is a variable set-up time between jobs. Set up may be faster if the same surgeon handles two patients in succession. Further dialogue between the two projects may be helpful.

On July 20, 1990, Davis visited the Departments of Computer Science, Electrical Engineering, and Management at the University of Vermont. The purpose of the visit was to gain information about state-of-the-art computer system applications which could benefit the Quick Response project. The Computer Science Department has one of the most advanced networks of workstations in the U.S., which links multiple departments. It is based on the increasingly popular X-windows system which we used in the Quick Response project. The network includes X-window terminals and IBM PS/2 machines running X-window. Users of any type of machine on the network can easily access files and applications on any other machine on the network.

The situation there suggested that our choice of the X-window system as a platform for the manufacturing scheduling system is a good one. The X-windows system is successfully used by novices as well as experts, and many people who formerly favored MS-DOS

On May 17, 1990, Professor Heimo H. Adelsberger of the Technical University of Denmark, Ms. Karen Biddle, Mr. Steve Freudenthal, Mr. Bill Edwards, Ms. Mary Alice Cooper, and Mr. Cary Grant of Milliken & Co. in Spartanburg, and Mr. Dieter Steinmann of the University of Saarland, Saarbrücken, Germany, met at the Clemson Apparel Research Center (CAR) to review the various projects which Kanet and Davis have underway.

On May 18, 1990, Mr. Hermann Havermann visited CAR. Mr. Havermann had donated a copy of his leitstand software product which we used for embedding some of our ideas regarding scheduling algorithms into his software, and tested in a reorder application as in Milliken's facilities.

June, 1990

On June 20, 1990, Kanet visited Micro Dynamics Corporation in Dallas, Texas, where he met with Dr. Ron Gordon to discuss possible cooperation with the Quick Response Planner project. Kanet gave a demonstration of the leitstand prototype that AHP Havermann had donated to Clemson University.

July, 1990

In July, 1990, Davis visited Mr. Christian P. Vogt, Vice President for Information Systems at the Medical Center Hospital of Vermont. Mr. Vogt was developing a large scale scheduling system for hospital operations. Davis and Vogt exchanged information about scheduling projects. Vogt's project and ours had a lot in common. For example, the Medical Center must efficiently schedule each critical resource such as an operating room. The room is analogous to a single machine. It processes jobs (operations on patients) and there is a variable set-up time between jobs. Set up may be faster if the same surgeon handles two patients in succession. Further dialogue between the two projects may be helpful.

On July 20, 1990, Davis visited the Departments of Computer Science, Electrical Engineering, and Management at the University of Vermont. The purpose of the visit was to gain information about state-of-the-art computer system applications which could benefit the Quick Response project. The Computer Science Department has one of the most advanced networks of workstations in the U.S., which links multiple departments. It is based on the increasingly popular X-windows system which we used in the Quick Response project. The network includes X-window terminals and IBM PS/2 machines running X-window. Users of any type of machine on the network can easily access files and applications on any other machine on the network.

The situation there suggested that our choice of the X-window system as a platform for the manufacturing scheduling system is a good one. The X-windows system is successfully used by novices as well as experts, and many people who formerly favored MS-DOS

applications on the IBM PS/2 have found the X-window system so convenient that they now use it most of the time. Therefore it seemed likely that the software we were developing under the X-windows system would be appropriate for use by non-technical people on the factory floor to schedule work, and a network could conveniently link scheduling systems in different parts of a firm.

Davis viewed the very high resolution workstations which were available on the network. Although the workstations are much more expensive than today's personal computers, in the near future low-cost personal computers will probably be available with graphics far superior to today's models. Our use of the X-windows system will facilitate future software modification to take advantage of improved graphics capability. (advanced graphics drivers may not be available until later for MS-DOS or OS/2 operating systems.)

August, 1990

On August 2, 1990, Kanet visited Mr. Bob Guise of Dun & Bradstreet Software in Atlanta, GA. While there, they discussed software design issues for state-of-the-art production scheduling systems.

On August 24, 1990, Kanet demonstrated the Quick Response Planning prototype and the AHP Leitstand to Mr. Bill Epstein, President of Iva Manufacturing in Iva, SC. Mr. Epstein expressed great interest in this work and we intended to keep him up to date as work progresses.

On August 31, 1990, Mr. Chuck Scheibe and Mr. Bob Joyce of AHP Software, Atlanta, GA, visited Clemson Apparel Research to review the Quick Response Planner project. Havermann & Partner, GmbH, Munich, Germany, had donated a copy of the leitstand system which we got up and running for demonstrations at CAR.

September, 1990

On September 17-21, 1990, Kanet attended a training course at Skokie, IL, in the object oriented programming language ACTOR. ACTOR appears to be a language very well suited to our needs. What makes it particularly attractive was the modest price and its ability to run under DOS. In order to run under DOS, Windows 3.0 must be present. But that was still not a major problem because of its modest price as well. In the Fall of 1990 we began efforts to experiment with the ACTOR language and more carefully assessing its suitability to our needs.

During September, 1990, Davis travelled to Burlington, VT, to visit members of the University of Vermont faculty and the Computer Center to get their help and ideas on implementing X-window applications. He also met representatives from DEC concerning graphical applications.

October, 1990

On October 29, 1990, Kanet organized a seminar entitled "Enterprise-Wide Data Modelling," presented by Professor Dr. A.-W. Scheer. Professor Scheer holds the chair of Business Administration and Information Systems at the Universität des Saarlandes, Saarbrücken, Germany. His research activities focus on EDP-systems for Computer Integrated Manufacturing (CIM), enterprise-wide data modelling, expert systems applications, and the development of general concepts for information processing. In his presentation Scheer argued that the nineties will be marked by significant changes in the design of industrial information systems. Traditional department- and function-oriented approaches will be replaced by integral approaches with the focus on the real-world situation of the enterprise rather than operational programming issues. The enterprise-wide data model can be utilized as a reference model for industrial companies and thus greatly reduce the costs of enterprise-wide data modelling while simultaneously providing all of its benefits.

November, 1990

On November 9, 1990, Kanet organized the seminar entitled "The Seven Myths of Finite Scheduling," presented by Mr. Robert F. Guise, Jr. Bob Guise is Senior Management Consultant with MSA Advanced Manufacturing, Inc., the world's largest provider of software for manufacturing, headquartered in Atlanta, GA. He has 35 years experience in manufacturing, integrally involved in the emergence of the computer as a powerful new tool in aiding manufacturers to be world-class competitors. In his lecture, Guise stated that the most important element of control in the typical manufacturing environment is the quality and timeliness of the decision-making procedures for assigning resources to various production goals, and customer delivery promises. Finite scheduling is the art and science of assigning manufacturing tasks to specific limited capacity resources.

Davis travelled to Atlanta on November 16, 1990, and November 27, 1990, to investigate algorithm animation research at Georgia Institute of Technology. He discussed animation capabilities with Prof. Stasko and viewed his software demonstrations. Algorithm animation is a graphical portrayal of the progress of the algorithm as it is being processed on a computer. For example, if the algorithm is a simple sorting routine which is being used to sort a list of numbers, the animation could represent the numbers being sorted as bars on a bar chart. As the algorithm progresses, the viewer can watch the bars being rearranged. If an algorithm is represented in object-oriented software, an animation could be constructed such that the viewer can observe changes in states of the objects and can see "envelopes" being sent between objects as messages are passed.

This animation can serve many different useful purposes. It can be used to help explain how an algorithm works. it could be used as an aid in modifying an algorithm such that it better suits user needs.

December, 1990

On December 7, 1990, Kanet organized the seminar entitled "Non-Chronological Scheduling," presented by Dr. Barry R. Fox. Dr. Fox works with the McDonnell Douglas Space Systems Co., Houston, TX. There he applies scheduling research to the exciting planning and scheduling problems in the space domain. His work there has been successful, resulting in a product known as COMPASS (Computer Aided Scheduling System). In his presentation Dr. Fox said that NASA faces scheduling problems with unprecedented scale and scope. It must develop powerful, generic scheduling capabilities that can be used in the implementation of a broad spectrum of space shuttle and space station scheduling applications. Reliance upon familiar chronological (simulation-based) scheduling methods may fail to satisfy important requirements or may result in exceptionally complex software. Non-chronological scheduling methods satisfy major requirements and provide a unified approach to the resolution of constraints and requirements.

February, 1991

On February 1, 1991, Kanet travelled to Iva Manufacturing in Iva, SC, to continue to search for ways in which the project team can transfer technology. Iva Manufacturing is one of the first companies in this area to install a Unit Production System and is a likely candidate for transfer of advanced technology like the Quick Response Planner.

On February 21, 1991, Kanet organized the seminar entitled "A Leitstand in a CIM-Concept," presented by Mr. Hermann Havermann. Mr. Havermann is the founder of the software house AHP Havermann & Partner, GmbH, Planegg, Germany. AHP specializes in modular solutions for shop floor control systems. In his lecture Mr. Havermann explained the philosophy behind the leitstand concept. The AHP-Leitstand is a finite scheduling system which receives released orders from an MRP II system and supports detailed planning with powerful planning tools. The leitstand is a real time system processing messages from the shop floor, providing visibility of the actual situation of the shop floor and reporting information back to the MRP II system. The AHP-Leitstand has been installed more than 300 times in different manufacturing branches and has produced excellent economic results. Future developments of the AHP-Leitstand will lead to knowledge based technologies.

March, 1991

On March 28, 1991, Kanet organized a seminar entitled "Real-Time Scheduling," presented by Dr. Richard Conway. Dr. Conway is Emerson

Electric Company Professor of Manufacturing at Cornell University. In 1967 Richard Conway Published Theory of Scheduling (with co-authors William L. Maxwell and Louis W. Miller), a book that essentially launched scheduling as an academic discipline. In his presentation, Conway argued that production planning and scheduling is on the threshold of a significant change in the mode of operation. Integrated, real-time, interactive scheduling is now feasible, and will offer some manufacturers a substantial competitive advantage. In addition to his lecture, Conway gave a demonstration of the "Production Reservation System" which he and Professor W. Maxwell developed. PRS has an order entry module, administration module, and a schedular module. Order entry can be done in an interactive fashion. Orders can be entered from the keyboard, with the parameters which are not specified taking on default values.

April, 1991

In April Kanet travelled to Saarbrücken, Germany, where he presented the paper "Anwendungssoftware für die Produktion in den USA - Aktueller Stand und Perspektiven für die 90-er Jahre." The conference was sponsored by the Ausschuss für Fertigung e.V. (The German equivalent to the American Production and Inventory Control Society). As part of his presentation Kanet outlined the goals and objectives of the DLA sponsored Clemson Apparel Research Center project and gave a synopsis of the Quick Response Planner project.

May, 1991

The importance of investigating intelligent help capabilities for manufacturing scheduling systems has been recognized by several prominent researchers. For example, Professor Heimo Adelsberger (Technical University Denmark) is participating in one of the leading research efforts as part of the European ESPRIT project. He visited Clemson University to describe the latest plans for developing an advanced "Leitstand" or scheduling system. Plans call for developing an "intelligent advisory" system. Since the scheduling system will include sophisticated routines for optimizing schedules, among other things the advisory system will help explain to the user why certain decisions have been made. For example, a user may be disturbed by the appearance of machine idle time in a schedule which has been automatically generated. But the intelligent advisory could provide a convincing rationale for this idle time and might thereby preclude the user from manually overriding the suggested schedule (and thus produce a worse schedule). Adelsberger admitted that the construction of the intelligent advisor would be very difficult and would need to take advantage of the latest research results in this area.

On May 28, 1991, Mr. Vic Angle, Director of Technology for Springs Industries, visited Clemson University to review the project in Computer Integrated Manufacturing Logistics. As part of the visit

Mr. Angle was given a demonstration of the Quick Response Planner as it has been implemented on the SUN SPARC workstation. Springs Industries is undergoing a major renovation in their technology in order to become more competitive. Mr. Angle expressed interest in a future co-operation between Clemson researchers and Springs.

June, 1991

During June a new version of the leitstand software was received from AHP Havermann & Partner. The software was installed along with installing a new Microsoft C 6.0 Compiler to provide a good OS/2 based development environment. Havermann has additionally promised to provide a set of libraries allowing quick access to a scheduling database. When this is received the developers environment can be updated so that experiments can be conducted in the AHP software environment.

On June 18, 1991 Kanet visited National Cash Register Corporation (NCR) in Liberty, South Carolina. NCR personnel has read about the Quick Response Planning System and were interested in receiving more information about leitstand based systems research.

July, 1991

Having read about the work in production planning being carried on at Clemson, Mr. Charles Fortune and Depak Chawla of AT&T Consumer Product Division contacted Clemson to learn more about our work in the area. On July 3, 1991 Mr. Fortune and Mr. Chawla visited Clemson and were given demonstrations of the Quick Response Planner and the AHP Leitstand.

On July 19, 1991, Mr. Vic Angle, Director of Technology, Cathy Harrison, Industrial Engineering, J. Earl Wood, Administration Manager, Chris W. Fischesser, Production Control Manager, from Springs Industries, visited with Kanet to get an update on Manufacturing Logistics research projects at Clemson. As part of the visit the group was given a demonstration of the Quick Response Planner (now ported to the SUN workstations in Riggs Hall, Clemson University) and the AHP Leitstand that was donated to Clemson by AHP Havermann. Initially the group was given a quick tour of the Clemson Apparel Research facility in Pendleton. Springs application is in the manufacturing of bedding (pillowcases, sheets, bedspreads, etc.) and was very interested in scheduling projects underway at Clemson.

On July 26, 1991, Kanet, Professor Peck, and Professor Sridharan visited Iva Manufacturing to meet with Mr. Bill Epstein, President of Iva Manufacturing. The purpose of the trip was to get a better understanding of the scheduling and planning procedure that a typical apparel manufacturer might be using and to see how the general leitstand model might fit the apparel scheduling problem. Iva schedules their work on a four week lead-time basis. Discreet

orders tend to "flow" through the factory because an individual order seizes the attention of all the operators in a work area as it goes through the shop. Based on this preliminary look there does appear to be application of leitstand software for Iva. Follow-up meetings to exploit this concept further are planned with Dr. Peck and Dr. Sridharan.

August, 1991

On August 27, 1991 Kanet organized the seminar entitled "Shop Floor Control: A New Approach For An Old Problem," presented by Professor Dr.-Ing. Hans-Peter Wiendahl from the University of Hannover, Germany. Professor Wiendahl is one of Germany's leading figures in the field of shop floor scheduling and control with numerous publications to his credit. His well-received book **Belastungsorientierte Fertigungssteuerung** (Load-Oriented Production Control) will soon be available in English. In his lecture Wiendahl described the conflict in shop floor control between short delivery time and good schedule performance on the one hand, low inventories and good utilization on the other hand. In Western Europe, but especially in Germany, numerous attempts have been made to solve this conflict. More than 100 commercial systems are available, but have often created disappointment. In the last few years a new concept, developed at the Institut für Fabrikanlagen, has been put into practice. It is based on the so called "funnel model." This lecture briefly described the underlying theory of this approach and outlined the experiences with two new software tools currently in use in German industry.